MechWarrior 5: Mercenaries Mod Editor Guide (v2.1)

*Entries in **bold** are new or have changed as of this guide version*

Contents

# Mod Editor Guide Changelog

v2.1 Guide Changes

- Added a 'Using Custom Data Tables' section covering how mods can now reference and use custom DataTables for Weapon and 'Mech Data Assets.
- Added a 'Using Custom Tag Containers' section covering how mods can now create and use custom Tag containers. This feature will be useful for creating new 'Mechs and Weapons without needing to modify any of the global tag containers.
- Added an 'Including External assets with your Mod' section covering how key external assets, such as WwiseAudio Content, should be organized prior to the packaging process.

Introduction

To access the MechWarrior 5: Mercenaries Mod Editor through the Epic Games Store, your Epic account must own the MechWarrior 5: Mercenaries game.

Use of the Mod Editor is subject to your agreement to the Mod Editor End User License Agreement (https://mw5mercs.com/static/docs/MW5_Mod_Editor_EULA.pdf).

As an owner of MechWarrior 5: Mercenaries, you will already have access to the MechWarrior 5 Editor package in your Epic Launcher Library. **If possible, we strongly recommend installing the Mod Editor onto an SSD.** We also **strongly recommend** that you set up some kind of version control or reliable backup system before beginning any serious Mod work.
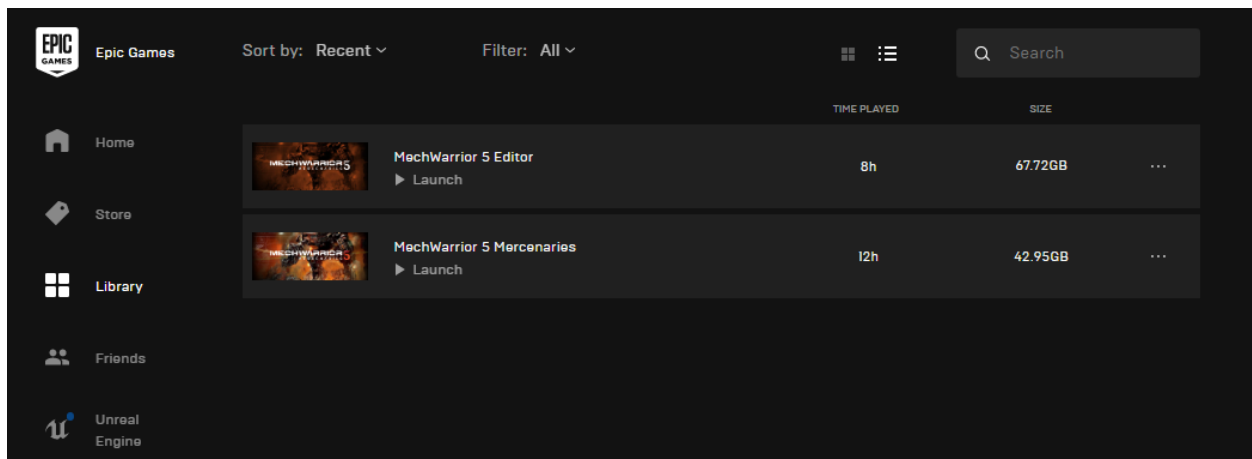
While a comprehensive rundown on Unreal Engine 4 is well beyond the scope of this initial guide, we will run through some of the basic information you'll need for becoming familiar with the Mod Editor. The final section of this guide will also cover what is required for making audio Mods for MW5 using the Wwise Authoring software from Audiokinetic.

By the end of this guide we will have covered:

- Creating a Mod

- Modifying an existing game asset in your Mod

- Adding a new asset to your Mod

- Packaging a Mod

- Creating a basic Test Map

- Spawning AI 'Mechs in a Test Map

- Spawning yourself as a 'Mech in a Test Map

- Creating a new 'Level 6' Weapon

- Modifying a 'Mech Loadout asset

- Setting up Wwise to support audio modding

Beyond this guide, there is a wealth of information and a huge community of UE4 developers out there to explore.
Here are just some of those great resources to get started:

- Unreal Engine 4 Documentation: https://docs.unrealengine.com/en-US/index.html

- Unreal Engine 4 Online Learning: https://www.unrealengine.com/en-US/onlinelearning

- Official 'Getting Started with UE4' Playlist on YouTube: https://www.youtube.com/user/UnrealDevelopmentKit/playlists?view=50&sort=dd&shelf_id=17

- Official 'Live Training' Playlist on YouTube: https://www.youtube.com/playlist?list=PLZlv_N0_O1ga0aV9jVqJgog0VWz1cLL5f

Initial Launch

On launching the Editor you may see a popup that the Editor is Discovering Asset Data. Let this process - and the subsequent process of Loading MechWarrior Assets - fully complete before you start doing any work.
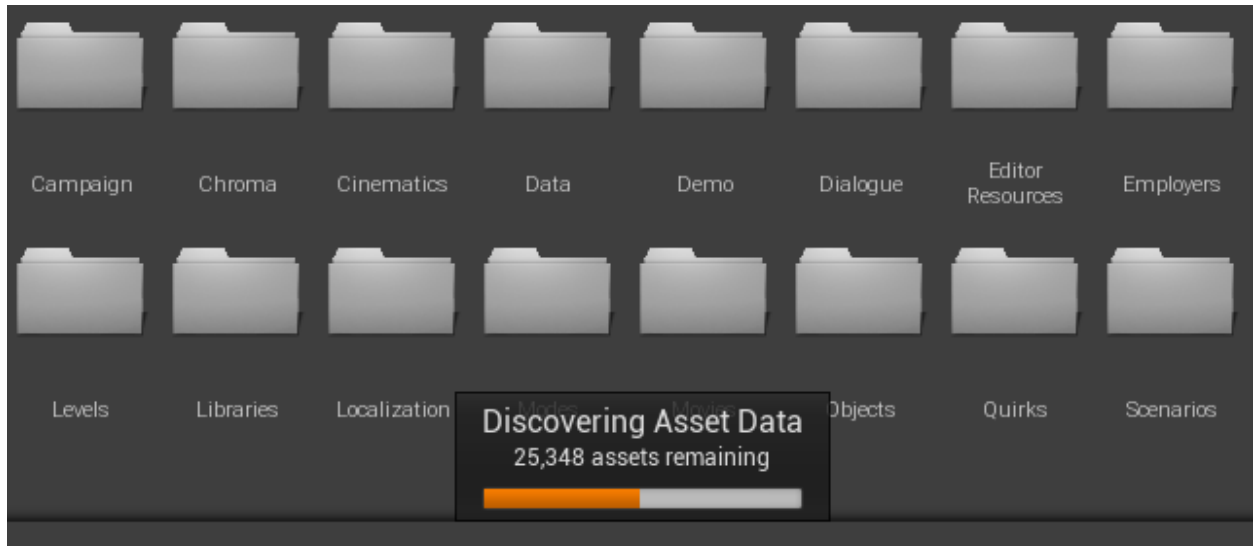
By default, the Editor will load you into a level called Title Screen. If you simply hit Play from here you'll launch into a basic version of the standard MechWarrior 5: Mercenaries title screen, from which you can do everything you can in the standard game, such as start a new Campaign, run Instant Action, and so on. The level used for the full Title Screen seen in the main game is called TitleScreen_Background, which you could open, modify, and play from if you wished.

Before we start modifying assets or creating new ones, we first need to make Mod package in which we will do our subsequent Mod work.

Making a Mod

As of v2, the Mod Editor features a more streamlined method for creating and packaging your Mods. In this section we'll run through the process for creating and packaging a Mod from start to finish.

After launching the Editor and allowing the asset loads to complete, you'll notice a series of modding-specific items at the end of the main toolbar along with the automatic popup of the 'Create Mod' interface.
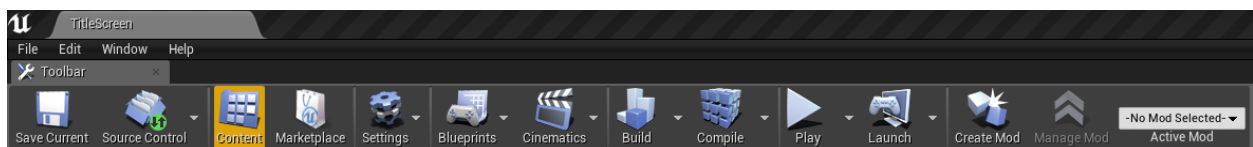
The 'Create Mod', 'Manage Mod', and 'Active Mod' features located at the end of the toolbar

Create Mod button

- Opens the Create Mod interface.
- Creating the Mod will build the necessary folder structure in which your Mod content will be located, and is the first step in any Mod project.

Manage Mod button

- Opens the Manage Mod interface for your Active Mod, through which you can package your completed Mod.
- The Manage Mod button will only be active if there is an Active Mod set.

Active Mod selector

- Used for setting the Mod project on which you are currently working.

Create Mod

The first step in starting your Mod project is to create the Mod package. This will build the necessary folder structure in which your Mod content will be located.

By default, the 'Create Mod' panel will appear when launching the Mod Editor. This is the same panel that appears when you click the 'Create Mod' button in the main toolbar.

You can stop the panel from appearing on launch by unchecking the 'Show on Startup' box at the bottom of the panel.
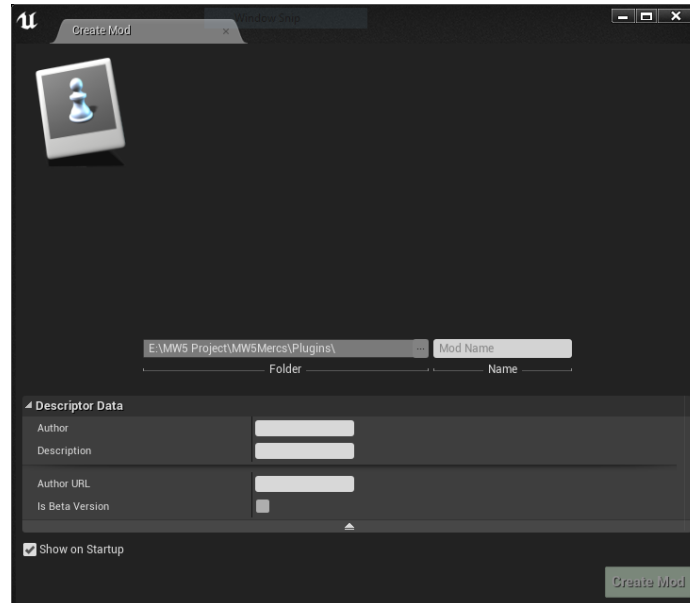
The only required field here is the Mod Name, but we recommend adding both an Author name and a Description.

For this example we'll call it TestMod.

Once you hit the 'Create Mod' button you'll see a message at the bottom right of the Editor stating the Mod was created successfully.
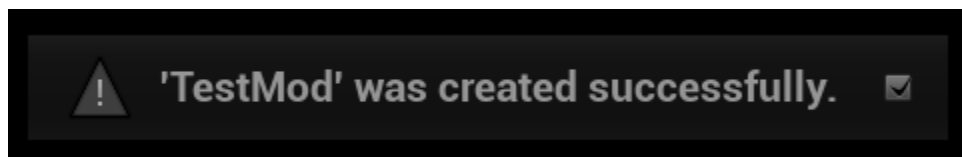
Now that you've created the Mod, there are some important things to note.

The first is a change to the main toolbar mentioned earlier in this section.

You can see that the 'Manage Mod' button is now available and that the 'Active Mod' is now set to the TestMod we just created.

Open a Content Browser window in the Editor if one isn't already visible. You can do so by selecting Content Browser from the Window dropdown.

In addition to the main Content folder, you should now also see two additional Mod-related folders.
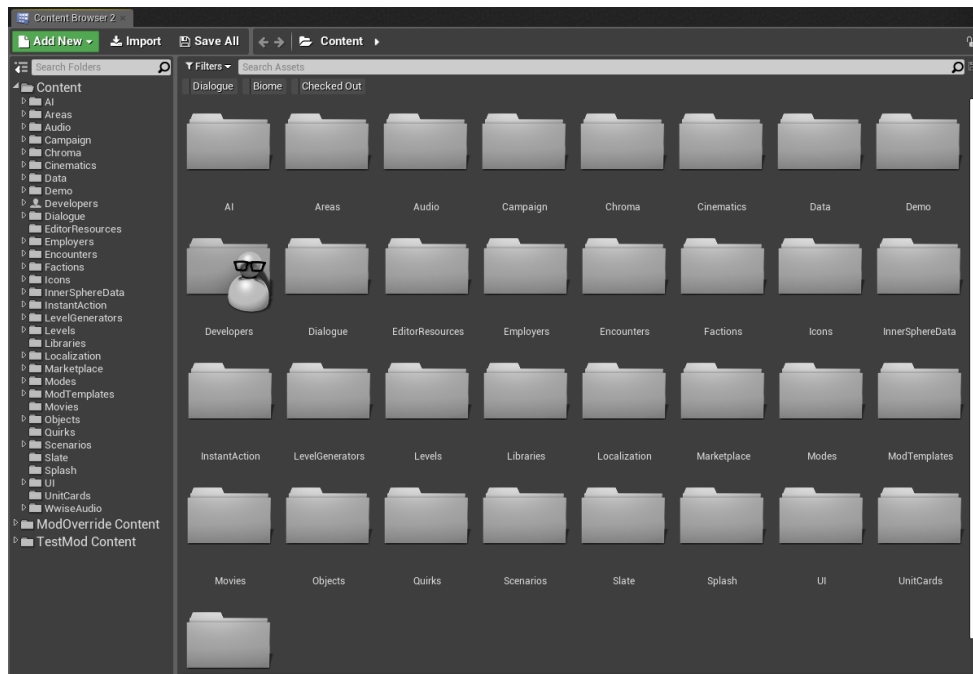
- ModOverride Content
  - Modded versions of existing game content (referred to in this doc as substitution Mods) will be located here.
- <ModName> Content
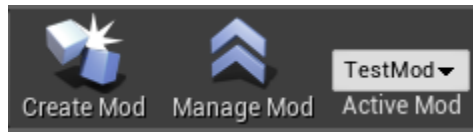 - New, non-substitution assets you've created for your Mod will be located here.

The contents of both the ModOverride and <ModName> Content folders are determined by your Active Mod.

This separation of substitution assets and new assets into these two folders is part of our solution for streamlining the creation of substitution Mods and ensuring the functionality of your Mods.

Creating a Substitution Mod

Let's start by making a substitution Mod (a Mod of existing game content). In this example we'll be modifying the stats of the Level 5 Gauss Rifle.

Navigate to the \Objects\Weapons\Gauss folder and open the GaussRifle_Lvl5 asset.

When looking at an existing game asset you'll see a new button in the toolbar called 'Save To Mod'.
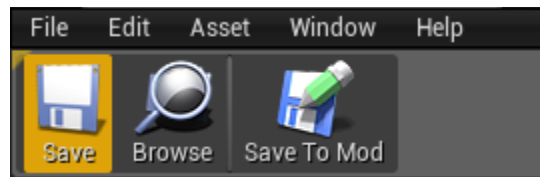

Clicking the 'Save To Mod' button inside an existing game asset will make a duplicate of the asset and save that duplicate to your ModOverride Content folder. Once a Mod version of a game asset has been saved, the Editor will automatically use that new Mod asset when playing; not the original.

This system allows you to safely modify existing game content without directly modifying the source assets.

With the GaussRifle_Lvl5 game asset open, go ahead and click that 'Save To Mod' button. The Editor will then seamlessly create a safe duplicate of this asset for modding, place it in the ModOverride Content folder for your active mod, open that moddable version of the asset, and close the original game asset.

When viewing the Mod version of an asset, its tab will show an additional Mod icon in addition to its standard asset type icon.

Additionally, the 'Open Game Asset' button will open the original game asset from which this Mod asset is derived if you wish to reference it.
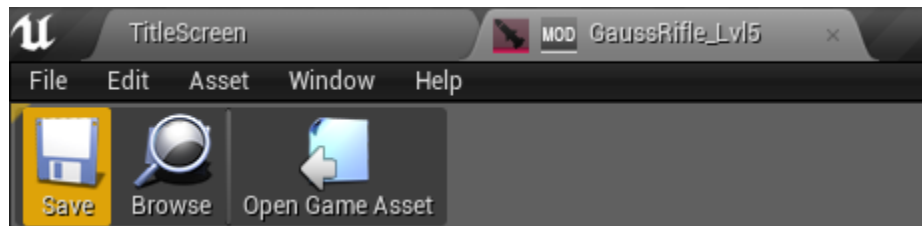
Before moving ahead let's quickly summarize the steps we've taken so far to create a substitution Mod:

1. Ensure you've created a Mod package and have it set as your Active Mod
2. Locate a game asset you want to modify and open it
3. Click the 'Save To Mod' button to create a safe version of the asset for modding
4. Make your changes inside that Modded asset

Now that we have our substitution Mod asset, let's modify some of its stats.

In this example we'll just increase the base damage and velocity.

Rather than modifying the source data inside the ProjectileWeaponStats data table, let's just uncheck the 'Read from Table on Save' box and make our changes directly in the asset.
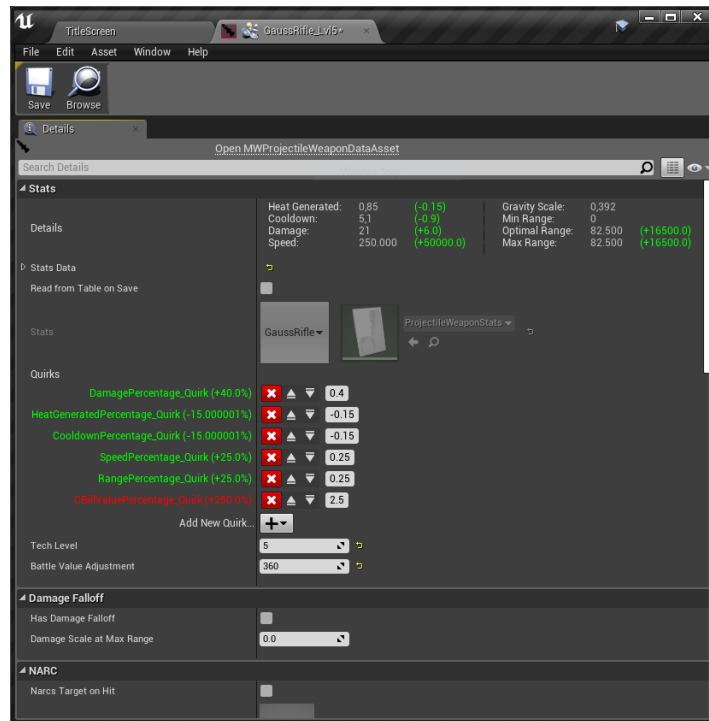
Now expand the Stats Data section and increase the Damage and Speed values, then save and close the asset.

We've now created our Modified version of the existing Level 5 Gauss Rifle.

The basic process for making substitution Mods will be the same for any other game asset. Let's reiterate those steps:

1. Ensure you've created a Mod package and have it set as your Active Mod
2. Locate a game asset you want to modify and open it
3. Click the 'Save To Mod' button to create a safe version of the asset for modding
4. Make your changes inside that modded asset

Creating a New Asset Mod

Before we proceed with packaging the Mod let's also create a new asset Mod. Rather than modifying an existing game asset we'll be creating an entirely new asset for the game to use.

For this example we're going to make a new Lance Formation asset.

As reference, you can find the current default Formation used by the game in the \AI_common\Formations folder. This is the layout of that default formation:
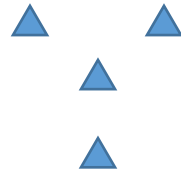
The player is located at the center of the above formation. We're going to make a new formation asset for the game to use instead that places the player at the tip of a more symmetrical diamond formation.
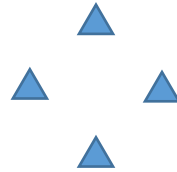
First, navigate to the \TestMod Content\Mods\TestMod folder.

Right-click inside that folder and choose the Formation asset option from the MW5 Misc category in the context menu. We'll call this new Formation asset DiamondFormation_Alternate. Open that newly created asset.
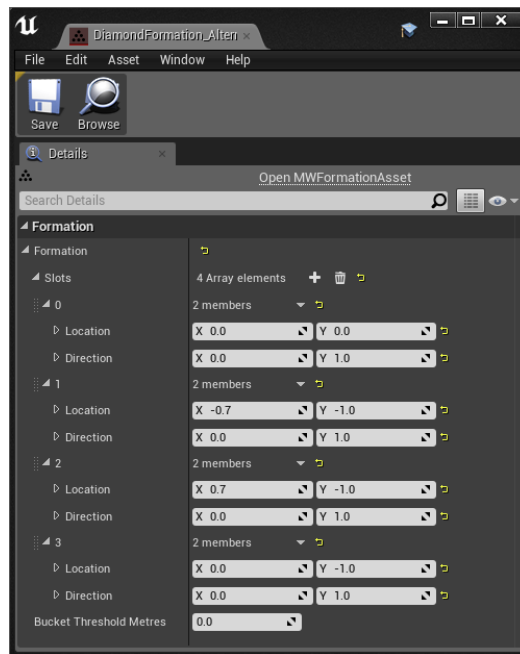
The settings shown in the above image will place the player (index 0) at the tip of the diamond formation.

Save and close that asset.

Now that we have our new Formation asset we need to tell the lance AI system to use it instead of the default Formation. The Formation to be used by your lance is determined by the SquadCommandComponent blueprint.

Navigate to the \Modes\MissionDirectorSystem\MissionComponents folder and open the SquadCommandComponent located there.

As we need to make a substitution Mod with this asset, click the 'Save to Mod' button. Then, click the 'Open Mod Asset' button.

In your new Mod version of the SquadCommandComponent blueprint, change the Default Formation Asset ID from the default DiamondFormation to your new DiamondFormation_Alternate asset. Save your Mod asset and close it.

Our Mod now contains three assets:

- A substitution Mod that changes the default stats of the Level 5 Gauss Rifle
  - *As a substitution Mod it is located in the ModOverride Content folder*
- A new asset Mod that introduces a new Lance Formation asset
  - *As a new asset Mod it is located in the TestMod Content folder*
- A substitution Mod that modifies the SquadCommandComponent to use our new Formation asset
  - *As a substitution Mod it is located in the ModOverride Content folder*

We are now ready to package our Mod using the 'Manage Mod' interface!

Manage Mod

When you're ready to package your Mod, click the 'Manage Mod' button in the main toolbar. This will open the Manage Mod interface.
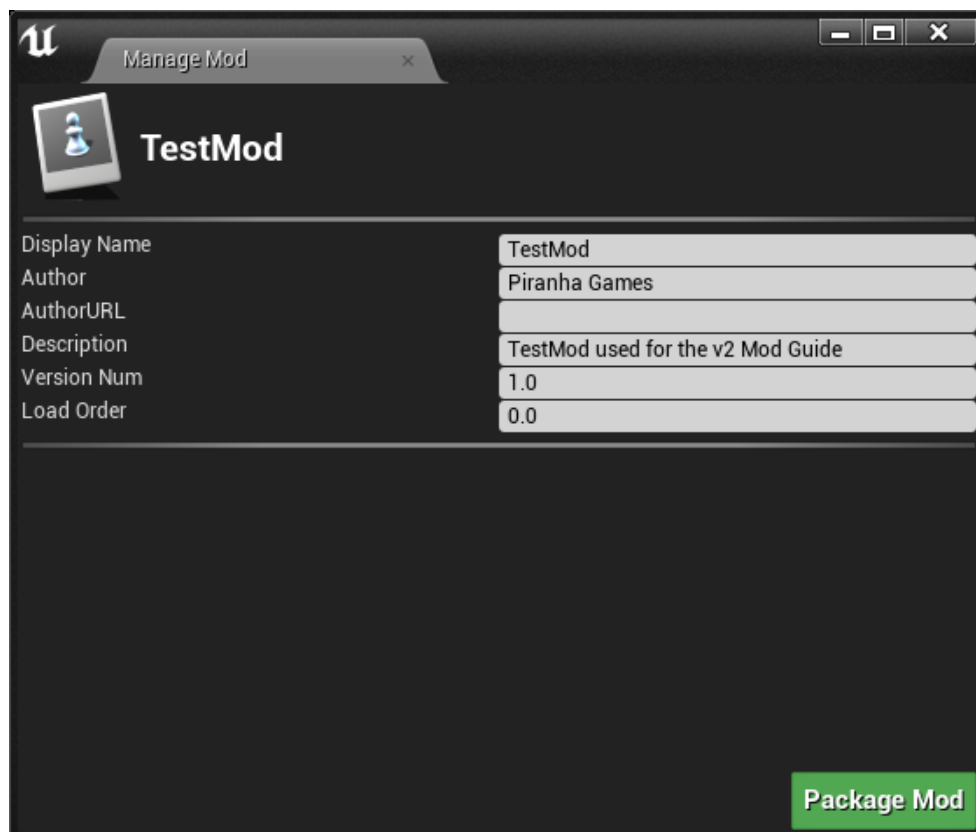
In this interface you'll see some of the details you set when the Mod was first created, in addition to two new fields:

- Version Num
  - The version number you want to set for this Mod. If you're packaging this Mod for the first time, a version number of 1.0 is a good starting point. If this is an updated version of a Mod you've already released, you'd set it to something like version 1.1 or 2.
- Load Order
  - The desired order in which this Mod will be loaded by the game.
  - Unless your Mod was explicitly made to accommodate or otherwise work in relation with another existing Mod, such as another weapon Mod, you'll want to keep the Load Order value at 0.
  - Users are free to change the load order of your Mod once they've downloaded it, enabling them to organize the load order of their installed Mods as they see fit.

For this example we're going to leave the Version Number and Load Order at their default values.

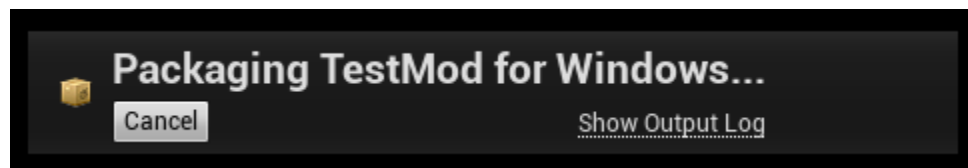When you're ready, click the 'Package Mod' button. You will then be asked to determine where the Mod package will be located. By default this will be your <Mod Editor>\MW5Mercs\Mods folder.

Click 'Select Folder' to confirm your desired location. In this example we're just using the default \Mods folder.

The packaging process will begin.

The Editor will tell you when the packaging process has completed.

Now that we've packaged the Mod, let's test it in-game. Close the Mod Editor and proceed to the next section.

Installing your new Mod to the game

Close your Mod Editor and launch MechWarrior 5: Mercenaries.

Select the Mods option from the Title Screen. Here you'll be able to view, enable, and disable your installed Mods.

If you don't already have a Mods folder in your game install, clicking the 'Open Mod Folder' will create one for you and take you directly to it in Windows Explorer. Click that button now.

Copy the folder for the Mod you created into this Mod directory, then return to the game.

Click the Refresh button in the Mods screen to update the list of installed Mods. Once the refresh is complete you'll see an entry for your recently installed Mod. Click the checkbox to activate it, then hit the Apply button.

You must restart your game for Mod changes to take effect.

After restarting your game, the Mod we just enabled will now be active!

Porting your existing Mod into the new Mod framework (Recommended approach)

If you've already made a Mod or have otherwise modified any game content with the previous version of the Mod Editor, porting your work into the new system is fairly straightforward.

1. Copy all of your modded content to a safe place outside the Mod Editor folder, such as a ModBackup folder in your main Documents folder. **This must be the pre-cooked version of the UASSET! Copy them straight out of the project folder structure, not your PAK files or cooking target folder.**
2. Update and launch the Mod Editor.
3. Click the Create Mod button and add information appropriate to the Mod you're porting, following the steps from the previous section on creating a Mod.
4. Your Project will now have a folder for new content and substitution content of the Mod you just created.
5. Open Windows Explorer and navigate to the folder where you placed your backup UASSETS from step 1.
6. Open another Windows Explorer window and navigate to the \MW5Mercs\Plugins folder of your Mod Editor. There you'll see a folder for the Mod you just created. Open that folder.
7. In here there are two folders we need to focus on:
   a. Content
   b. ModOverride
8. The Content folder is where you will place any UASSETS that are new, in the sense that they don't modify an existing game asset. For example, adding an entirely new Level 6 Gauss Rifle Weapon Asset.
9. The ModOverride folder is where you will place any UASSETS that are modified versions of existing game assets. For example, altering the values of the existing Gauss Rifle Weapon Asset.
   a. IMPORTANT: When placing assets in the ModOverride folder, they must have the same folder structure as the original asset. For example, if I've modified the basic Gauss Rifle Weapon Data Asset, I would place that modified UASSET in the following folder structure:
      - ModOverride
        - Objects
          - Weapons
            - Gauss
10. Launch the Mod Editor and you will see your respective assets in the Content and ModOverride folders of your Mod.
11. You've now ported your Mod content into the new system! To package your Mod in preparation for distribution, just run through the packaging process outlined in the Manage Mod section of this guide.

Porting your existing Mod into the new Mod framework (Basic approach)

Mods that have already been packaged and released to players using a previous version of the Mod Editor will still function, but they will not be supported by the new in-game Mods manager screen by

default. If for now you are just interested in updating your already-released Mod to work with the new in-game Mods management screen, the approach outlined here will allow you to do so without needing to perform the full conversion steps outlined in the previous section. All you'll need to do is make an empty Mod using the updated Mod Editor and place your existing PAK file in an appropriate folder. The steps below outline this process.

1. Click the Create Mod button and add information appropriate to the Mod you're porting, following the steps from the previous section on creating a Mod.
2. Your Project will now have a folder for new content and substitution content of the Mod you just created. We aren't going to be adding anything to the Mod in the Editor itself. We just needed to create the Mod as a container for your existing PAK file.
3. Package this Mod in the Manage Mod screen.
4. When packaging is complete the Mod Editor will ask you if you want to navigate to the folder where the Mod was created. Say yes to automatically open a Windows Explorer window to that location. Otherwise, you can find it in the \MW5Mercs\Plugins folder of your Mod Editor
5. Inside the folder for the Mod you just created, create a Paks folder.
6. Drop your existing PAK file into the Paks folder.
7. You're done! When a user receives this new Mod folder, dropping it into their game will make it compatible with the new in-game Mods manger screen.

Compatibility of Mods made in v1 of the Mod Editor

While the v2 update to the Mod Editor has introduced a streamlined method for Mod creators to create and package their Mods, Mods that were manually packaged by their author using a previous version of the Mod Editor can still be used in the game.

If a Mod you've downloaded was built in a previous version of the Mod Editor and only contains a basic PAK file, you can still use that Mod by placing it inside the \MWMercs\Content\Paks folder of your game install. The game will still read this data and implement the Mod, but it will not appear in the in-game Mods screen unless the author has ported their Mod using either of the Recommended or Basic options outlined above.

Changing the Load Order of a Mod after it has been packaged

While the Load Order of a Mod can be set when it is packaged, users can also manually change the Load Order of a downloaded Mod simply by modifying the .JSON text file that is included with Mods packaged in v2 of the Mod Editor.

To change the Load Order of a Mod, navigate to the root folder of the Mod in the \MW5Mercs\Mods folder of your game install.

Open the mod.json file with a text editor such as Notepad.

Locate the "loadOrder" line and change the listed value to your desired load order number.

Save and close the file.

The game will now load the Mod in whatever order you set!

Using Custom Data Tables

As of the latest patch Weapon Data Assets can now reference any specified DataTable, instead of being restricted to using the default stats tables.

As a result, different Weapon Mods no longer need to reference the same source table. This is a benefit particularly for mods that focus on modifying a specific collection of weapon classes or are introducing entirely new collections of weapons.

As long as mod makers are using their own stats tables, this would allow a player to install multiple Weapon Mods from multiple authors with minimal conflict, as each Mod would be deriving its weapon stats from a unique table, rather than all Mods directly modifying the default stats table.

To modify Weapon Stats at the DataTable level you need to approach your Mod as a new asset Mod, not an override Mod. Rather than modifying the existing stats table, you'll instead make your own and tell any desired Weapon Assets to use that table as the source of its base stats.

This means that instead of opening the existing ProjectileWeaponStats DataTable and using the Save To Mod feature, you would instead manually copy that DataTable into your Mod Content folder (not the ModOverride folder), give it a distinctive name such as ProjectileWeaponStats_Mod, and make your changes to that new copy.

Here are the basic steps, using our old pal the Gauss Rifle as an example asset.

1. Navigate to the ProjectileWeaponStats asset, located in the Objects/Weapons/_common/Config folder
   a. The Missile and Trace weapon stat assets are also located here
2. Drag the ProjectileWeaponStats asset into your <Modname> Content folder (not the ModOverride Content folder) and select the Copy Here option. This will add a duplicate of the Projectile stats asset to the new asset folder of your Mod.
3. Navigate to the <ModName> Content folder and rename the Projectile stat asset you just copied. ProjectileWeaponStats_Mod is a good name for it.
4. Right-click and select 'Export as JSON' to create an external source file for this DataTable (otherwise it will be trying to reference the original). Save it in the same location as your new ProjectileWeaponStats_Mod asset.
5. Now we want to Mod the existing Weapon Data Asset for the base-level Gauss Rifle to use this new DataTable. In the main content area, navigate to Objects/Weapons/Gauss.
6. Open the GaussRifle Weapon Data Asset and click the Save To Mod button. This will make an override duplicate of that asset.
7. In that asset, note the Stats dropdown is set to use the default ProjectileWeaponStats asset. Click that Stats dropdown and select our new ProjectileWeaponStats_Mod asset from the list instead.
8. Our modded version of the Gauss Rifle now gets its base stats from our modded version of the ProjectileWeaponStats file!
   a. Note that making changes to the stats file while an associated Weapon Data Asset is open may not cause the stats displayed in that WDA to visually update. You may need to close and re-open that WDA if you want to confirm your changes are reflected in the displayed stats.

Using Custom Tag Containers

As of the latest patch, custom Gameplay Tag files can be created and read by the Editor.

When a new Mod is created, the Editor will automatically create a Config folder in the associated Mod folder.

> <Editor Install Location>/MW5Mercs/Plugins/<ModName>/Config

Within that Config folder you'll find a Tags folder, within which will be located an INI file named <ModName>Tags.ini.

Any Custom Tags you wish the Editor to use can be added directly to this file without needing to modify any of the source Tag assets. This will allow you to add things like new 'Mechs and new Weapons without introducing conflicts with other Mods that may also be adding new Tags to the game.

When you run the packaging process through the Manage Mod interface of the Mod Editor it will automatically package this Config folder for you.

If you already created Mods prior to the last update, you can run through the steps below to add this mod-specific tag file to your existing Mods.

1. In Windows Explorer, navigate to <Editor Install Location>/MW5Mercs/Content/ModTemplates/BaseTemplate
2. Copy the Config folder located there and paste it into your desired Plugins/Mod folder. For example, <Editor Install Location>/MW5Mercs/Plugins/<ModName>/
3. Open the Tags folder located in the Config folder you just brought into your Mod
4. Rename the PLUGIN_NAMETags file to match the name of your Mod. For example, GaussModTags

The Editor will now read this Tag file correctly when working with an existing Mod.

Including External Assets with your Mod

As outlined in the final section of this Guide ('Important steps before packaging your Audio Mod'), certain Mods require the inclusion of external assets that are not visible within the Editor itself. Some of the more common cases of this are the WwiseAudio files and Gameplay Tag INI files.

When packaging through the Mod Manager interface, the packager will now automatically include any files located in the /Plugins/<ModName>/Config or /Plugins/<ModName>/ModOverride/WwiseAudio folders of your Mod.

When making a Mod that relies on either WwiseAudio data or a Gameplay Tag file, ensure you have the necessary files located in their respective locations before packaging your Mod.
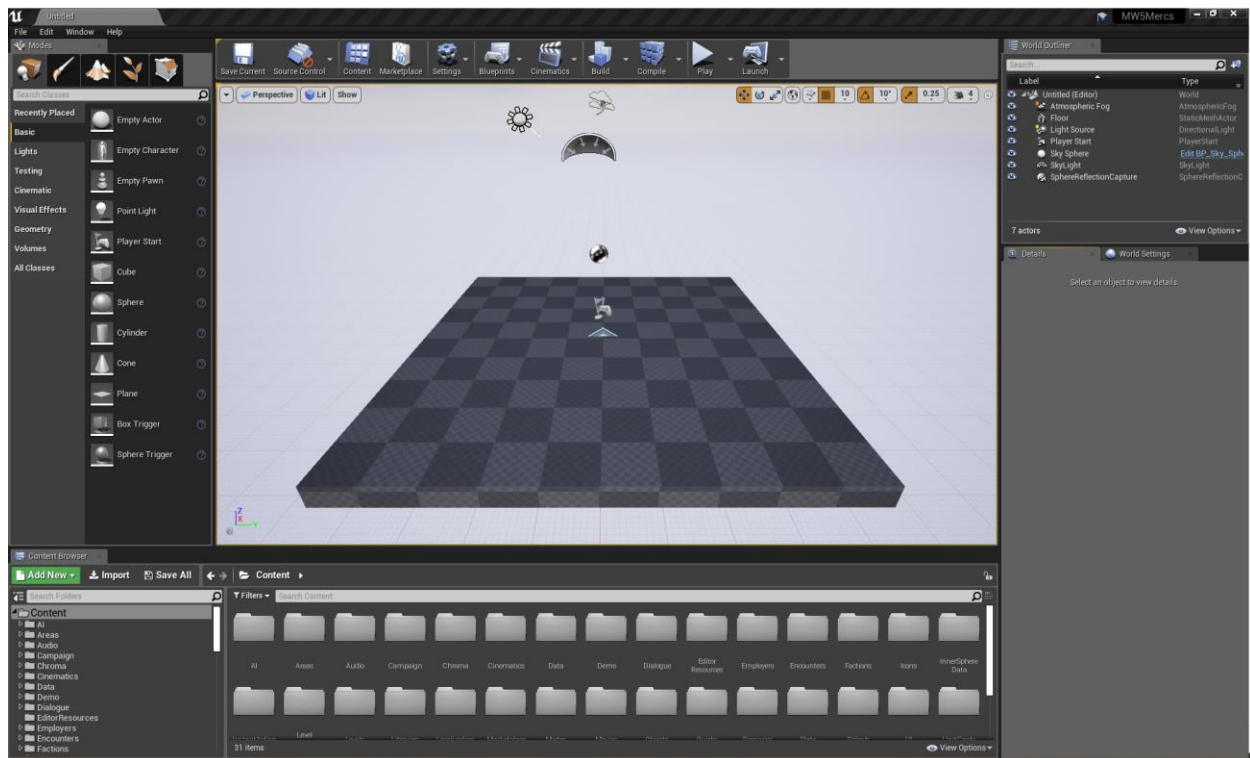
Using the Mod Editor

Creating a basic Test Map

When building any kind of gameplay Mod, such as a new Weapon or 'Mech variant, having a basic test level is a very useful tool for quickly testing your changes.

Hit CTRL+N on your keyboard to bring up the New Level panel (or select New Level... from the File option in the toolbar).

Choose the Default option.

This level contains some very basic elements:

- An Atmospheric Fog actor

- A Floor mesh

- A Directional Light

- A Player Start actor

- The default Unreal Engine Sky Sphere

- A Skylight

- A SphereReflectionCapture actor

The first thing you'll want to do is increase the size of the Floor mesh. This will give you a lot of room to play around in once you get into a 'Mech or start adding assets to the test level.

- Select the Floor actor in the World Outliner

- In the Scale field, change the X and Y size from 1.0 to something larger, such as 500.0.
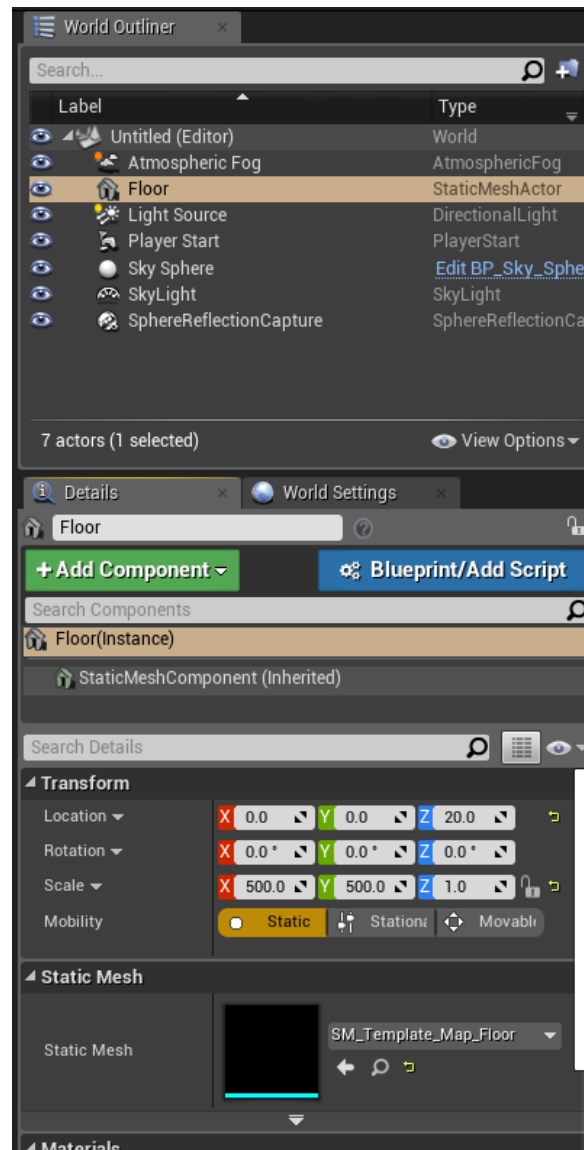
Now that you've got a bigger playing field, go ahead and save this level.

With your new level ready to go, let's drop a 'Mech Spawner down.

Spawning an AI 'Mech

In your Content Browser, navigate to Objects\Spawners. In that folder you'll see an asset called SpawnPoint.

Click and drag that asset into your level.

Select that asset inside the level or in the World Outliner pane.

In the Details pane you'll see some key variables:

- Spawn On Begin Play

    - Checking this box will cause the 'Mech to spawn when you Play or Simulate the level

- Spawner
    - From this dropdown you can choose from the various spawner types
- TeamID
    - This will determine whether the spawned actor is friend or foe
        - 0=Ally
        - 1=Enemy

For this example, set the SpawnPoint to Spawn On Begin Play, set the TeamID to 0, and set the Spawner to Spawner Mech
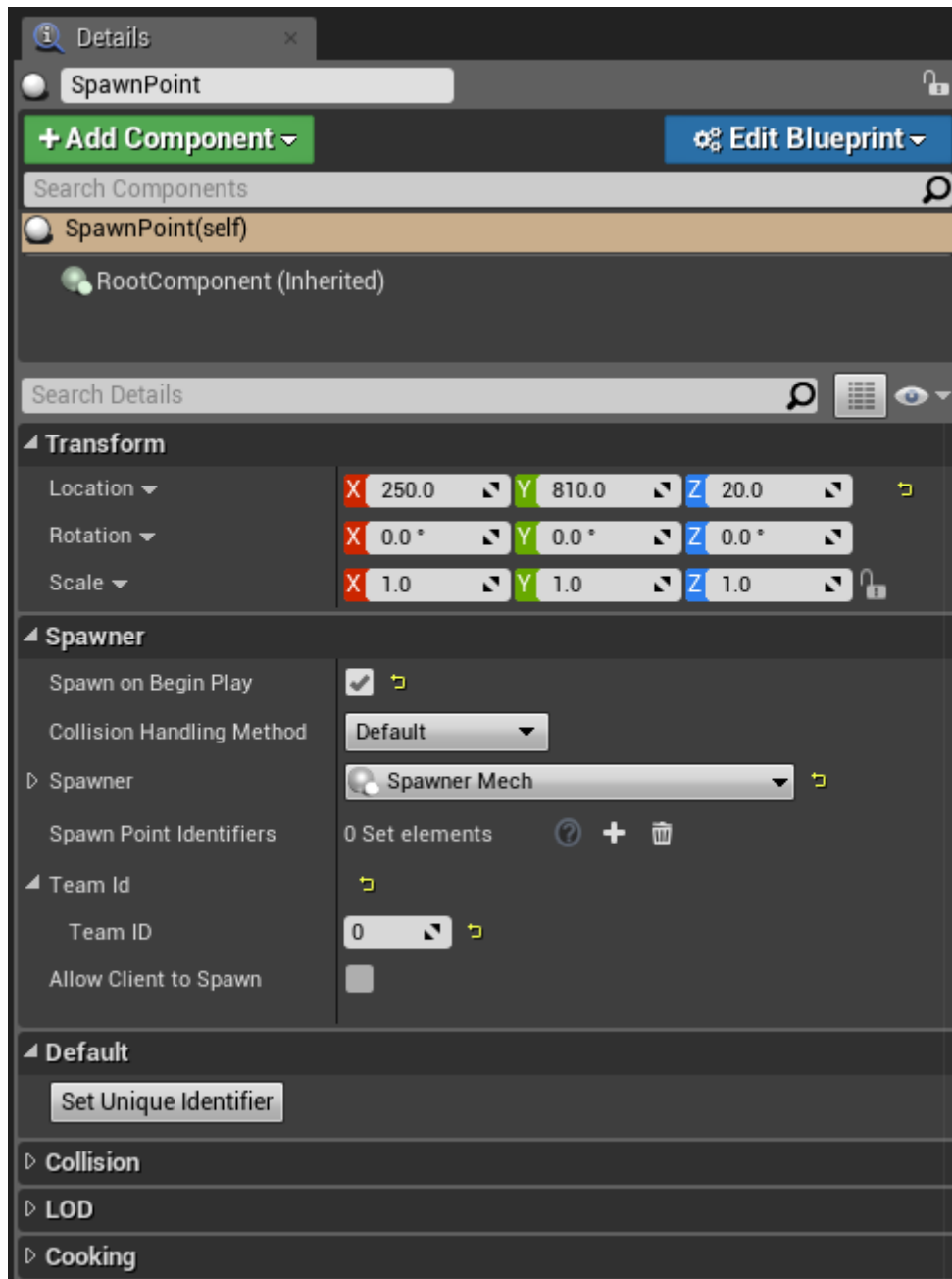
Now that you've set it to use a Spawner Mech you can expand that section to reveal additional details.
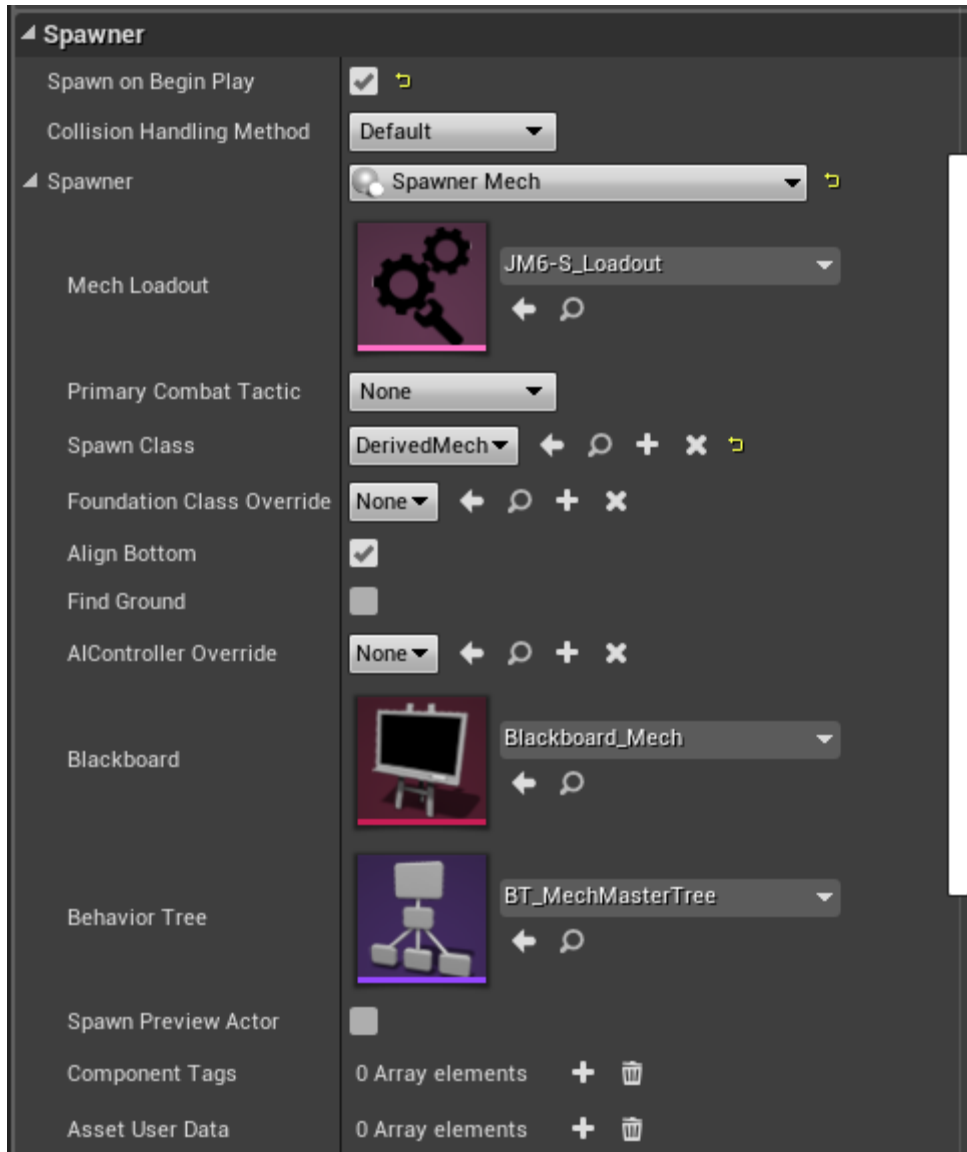
By default you'll see the spawner is set to use a JagerMech JM6-S. If you hit Play in the Editor now, you'll spawn in as a freefloating pawn and the JagerMech will spawn in the world. Hitting Escape on your keyboard will stop the scene and return Editor control. Try that now!

So that's how you can spawn a 'Mech into a test level. Simply drop a SpawnPoint, set the Spawner type, and set the Loadout/variant you want to spawn.

Now let's get you into a 'Mech.

Spawning as a 'Mech

Using the same level with that same AI SpawnPoint intact, select the World Settings option from the Window dropdown in the toolbar.

In the World Settings tab you'll see a GameMode Override dropdown. It will be set to 'None' by default. Select TestMode from that dropdown.

Now hit Play again, and this time, you'll spawn in control of a 'Mech.

If you set the SpawnPoint you placed in the previous step to use TeamID 0, the two of you will be on the same team. If not, get ready to throw down!

If you want to change the 'Mech in which you're spawning, click the little magnifying glass next to the TestMode dropdown in the World Settings pane. This will point your Content Browser directly to that TestMode asset.
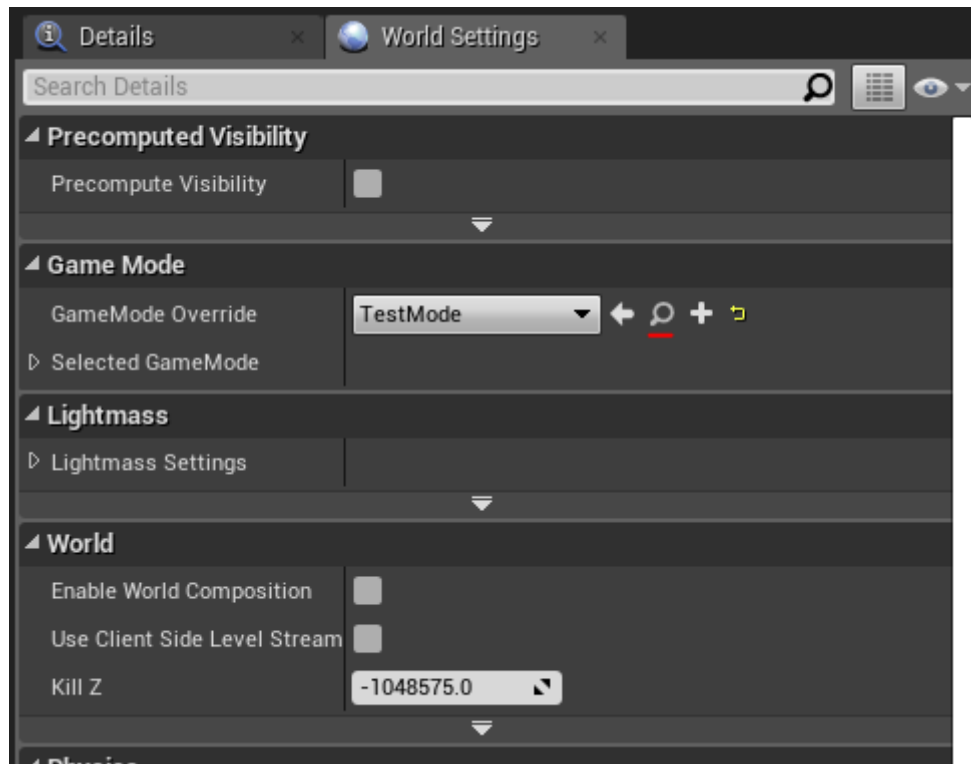
Double-click that TestMode asset to open it.

In there you'll see a lot of things, but for now just focus on the Campaign section of the Details pane. Expand the Starting Mechs section located there.

Details

Search Details

Actor Tick

Start with Tick Enabled	☑
Tick Interval (secs)	0.0
Allow Tick Before Begin Play	☑

Default

Spawn Count	0
Starting Pilots	0 Array elements ✚ 🗑
Starting Codex Entries	0 Array elements ✚ 🗑

Classes

Game Session Class	MWGameSession ▾ ← 🔍 ✕
Game State Class	DefaultGameState ▾ ← 🔍 ✚
Player Controller Class	MW5PlayerController ▾ ← 🔍 ✚ ↺
Player State Class	MWPlayerState ▾ ← 🔍 ✚
HUD Class	HUD ▾ ← 🔍 ✚ ✕
Default Pawn Class	DerivedMech ▾ ← 🔍 ✚ ✕ ↺
Spectator Class	MWFreeCameraPawn ▾ ← 🔍 ✚
Replay Spectator Player Controller Clas	MW5PlayerController ▾ ← 🔍 ✚
Server Stat Replicator Class	ServerStatReplicator ▾ ← 🔍

Game

Default Player Name	▾

Campaign

Startup Scenario	None — None ▾ ← 🔍
Starting Missions	0 Array elements ✚ 🗑
▲ Starting Mechs	2 Array elements ✚ 🗑 ↺
0	MWMechLoadoutAsset:ANH-1E_Loadout ▾ ← 🔍 ✕ ↺
1	MWMechLoadoutAsset:FLE-15_Loadout ▾ ← 🔍 ✕ ↺
Starting CBills	500000
Starting System Id	3445

Mech Warrior

Mech Class	DerivedMech ▾ ← 🔍 ✚ ✕
Default Transition Class	FadeTransition ▾ ← 🔍 ✚ ✕
Engagement Tracker	None ▾ 🔍 ✏
Default Inner Sphere Class	StarSystemGenerator ▾ ← 🔍 ✚ ✕

Click the ANH-1E dropdown to see a list of all the 'Mech Loadouts in the game. Simply choose the Loadout you want, go back to your main Editor window, and hit Play. You'll now spawn as the 'Mech you selected.

Let's go ahead and make a new Weapon!

Creating a new Weapon

In your Content Browser, navigate to \Objects\Weapons.

Here you'll find folders for all the main weapon classes. Let's create a new AC20.

Open the AC20 folder.

In here you'll see what are called Weapon Data Assets.

Right-click on the one that says Autocannon20_Lvl5, and from the context menu, select Duplicate.

Unless you gave it a custom name, your new asset should be called Autocannon20_Lvl6. As this is a new asset Mod, move your new Level 6 AC20 asset from the main Weapons folder and into your <ModName> Content folder.

Double-click the asset to open it.

For this guide we're just going to make some minor changes.

With the asset open you'll see some key areas for changing the default values for this weapon.
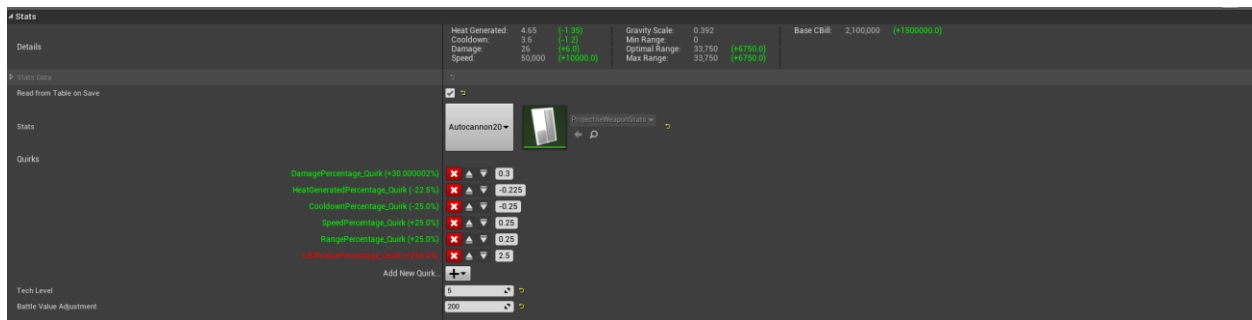
You can see here how Weapons have their base stats modified by Quirk Assets.

AC20s are kind of slow, so let's just change the value for the SpeedPercentage_Quirk from a 25% boost to 250%.

On second thought, let's also increase the CooldownPercentage_Quirk to -250%, because why not.

Look upon our works, ye mighty, and despair.

Now that we've made this new Weapon Asset, we need to add it to a Loadout.

Modifying a 'Mech Loadout asset

Go back to the TestMode asset you opened earlier. In the first selected 'Mech slot, set the 'Mech to the KGC-000 Loadout. Now, click the magnifying glass to see that asset in the Content Browser.
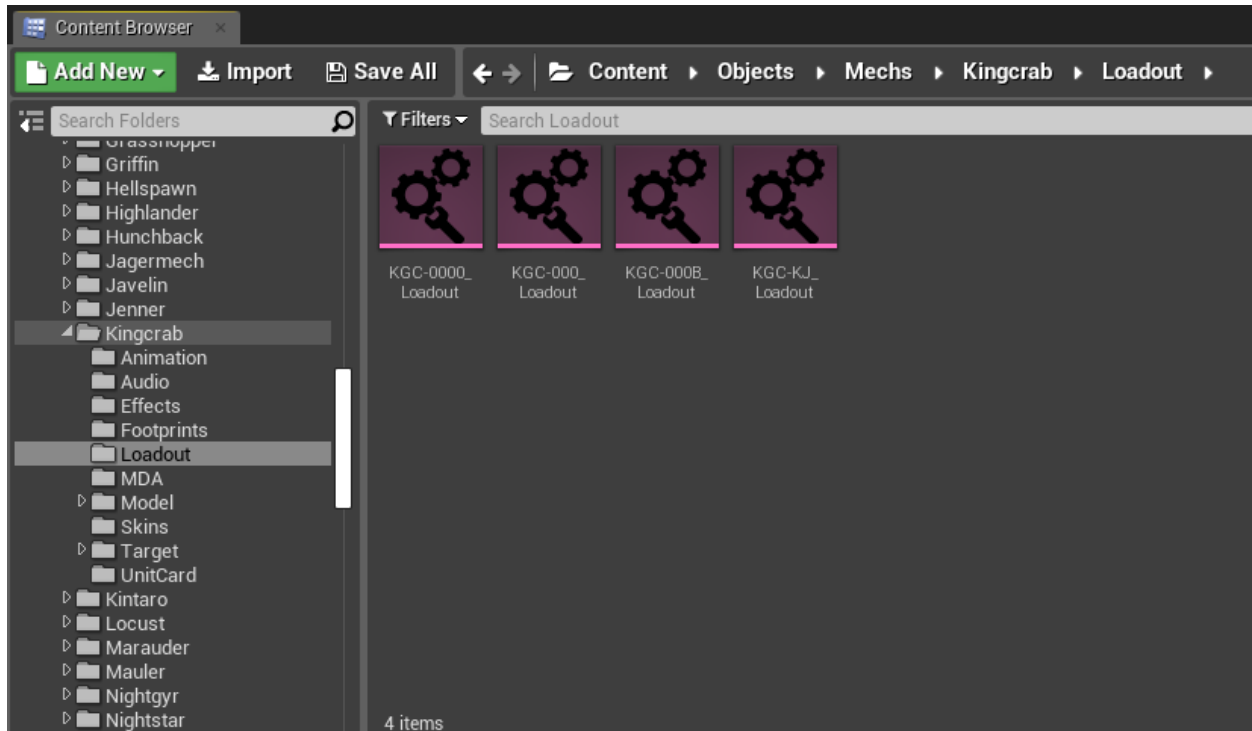
Rather than modifying the KGC-000 Loadout asset directly, open it and click the Save To Mod button. Then, click the Open Mod Asset button to open the new modded version.

In your new Mod asset, expand the Mech Loadout Section. Here you can see all the various attributes of the 'Mech in question. For now we just care about giving it that AC20 we made, so scroll down until you reach the Weapon Loadout section.
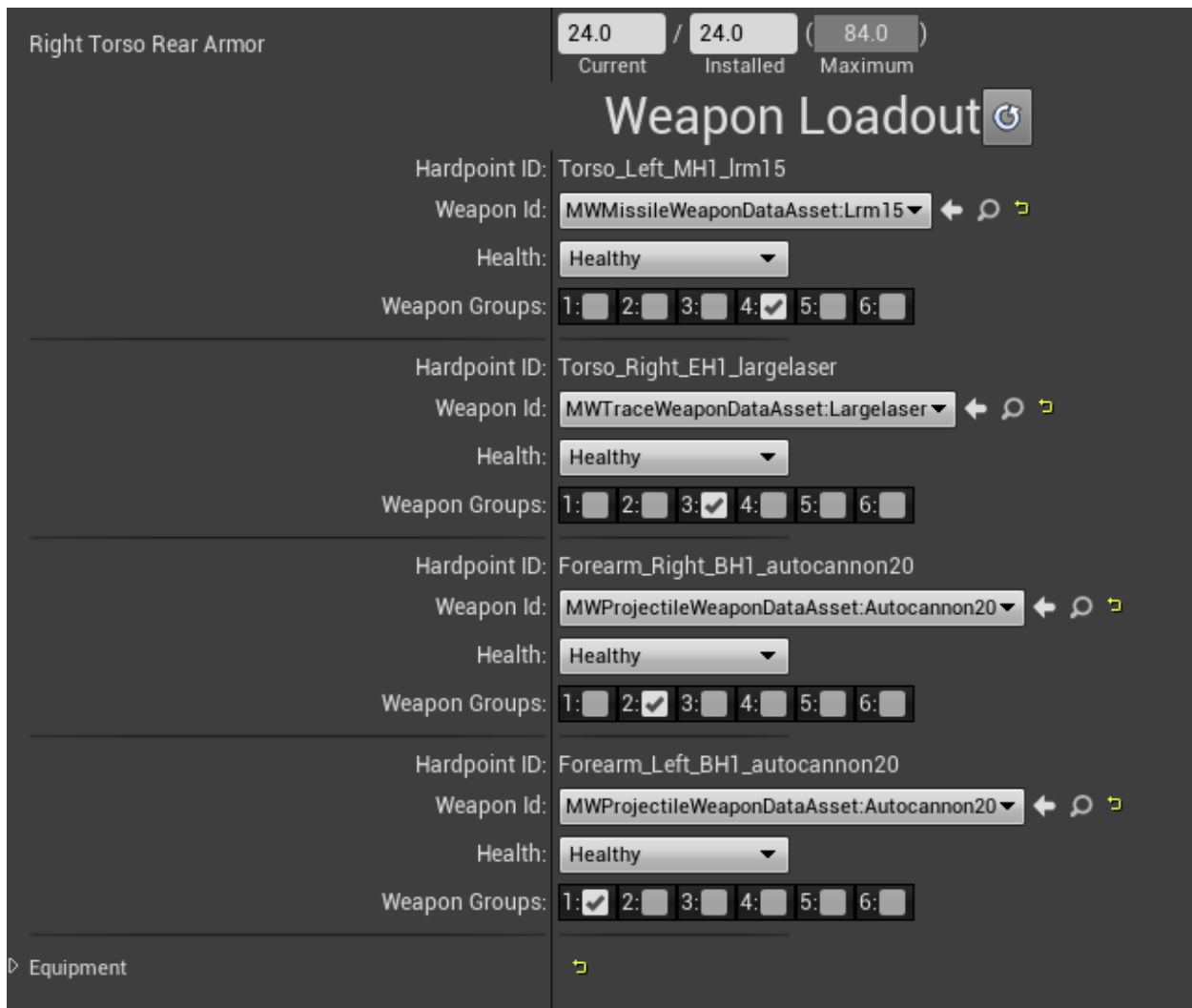
In the Autocannon20 dropdowns, select your new Autocannon20_Lvl6 asset and save this KGC Loadout asset.

Now that you've made and modified this new version of the KGC-000, go back to the TestMode asset you opened earlier and change the selected Loadout to the Mod version of the KGC-000.

Return to the main Editor window and hit Play. If you did everything right, you should spawn in the custom KGC-000 'Mech with your two completely ridiculous AC20s equipped.

Now go ahead and click the left mouse button to fire an AC20. Go on.

Good stuff.

Now that you know how to make a Test Level, spawn an AI 'Mech, spawn yourself in a 'Mech, create and modify a weapon, and modify a 'Mech Loadout, you should have a very basic understanding of how you can start modding assets in the MW5 Mod Editor.

As you comb through the Content Browser you'll see just how much stuff is there for you, just waiting to be modded. Campaign Arcs, Audio, Factions, UI, and so much more.

We're excited to see what you're going to make.

Using Audiokinetic Wwise to Mod music and sound in MW5

MechWarrior 5: Mercenaries uses Audiokinetic's Wwise for its audio framework. As a result, if you intend to make any audio Mods for MW5 you will need to download the Wwise Authoring software and acquire a non-commercial license from Audiokinetic. Fortunately, Audiokinetic provide **free** use of their Wwise Authoring software for non-commercial purposes!

Important Information

- While we will be working closely with Wwise to ensure support for your creation of MW5 audio Mods, the approval and continued availability of your non-commercial Wwise Authoring license is at the sole the discretion of Wwise. We cannot guarantee the provision of a non-commercial license for your Mod project. In the event that you are not granted a license - or if a provisioned license is revoked for any reason - we cannot provide you with an alternate license.

- If granted, Wwise's Non-Commercial License will be provided to you by Wwise on a quarterly basis. You will need to request an extension of the License from Wwise if you require additional time before releasing your Mod, or if you wish to continue working on and supporting your Mod after its release.

- Use of the Wwise Authoring tool is subject to the Wwise End User License Agreement.

- We **strongly recommend** that you set up some kind of version control or reliable backup system before beginning any serious Mod work.

Registering your Wwise Project

Registering your Mod Project with Wwise to acquire a non-commercial license is very easy. However, pay careful attention to the following section, as you will need to fill out the registration forms with specific information.

Create and validate your new Wwise account (https://www.audiokinetic.com/sign-up/)

Register your project (https://www.audiokinetic.com/register-project/)

On the initial registration screen, make these two selections:

- *Select the type of project you would like to register. This will help us provide you with the perfect solution for you and your organization.*

    - Non-Commercial

- *Please select the option that matches your project:*

    - I need more than 200 media assets

On the next screen you will be asked to enter in some details regarding your Mod Project. Fill them out according to the information below.

Your Project

- Project Name

  - MW5Mercs Modding (A Unique Name)

    - Your Project name **must** start with MW5Mercs Modding

    - Your Project name **must** include a unique name by which to identify it

    - Your Project name should be as specific and unique as possible. Using your online handle is a good way to ensure this, but we request that you also include additional information about the nature of your work

    - For Example: "MW5Mercs Modding (NewtonvsLeibniz's Sound Packs)"

- Project Website

  - Your website, if applicable

- Audiokinetic Contact

  - I don't have a contact yet

- Project Description

  - Enter a brief summary of your goals with when it comes to modding audio in MechWarrior 5: Mercenaries

- Pre-Production End Date

  - This should just be the date on which you've submitted your registration.

- Estimated Release Date

  - The estimated date of release for your Mod.

- Game Engine

  - Unreal Engine

- Specify Version

  - 4.23.1

- Wwise Integration Version

  - Other/Custom

  - 2019.1.4.7065

- Target Platforms

  - Windows

On the next page you will be asked to enter in your Company Details. In this case, treat the "Company" as yourself and fill out the details on this page according to the information below.

Your Company

- Company Name

    - Your full name

- Website

    - Your website, if applicable

- Company Size

    - The approximate number of people who will be working on this Mod

- Street Address, City, etc

    - Your home address

On the next page you will be asked to agree to the Wwise License Agreement. You must read and accept this agreement in order to proceed.

On the final page you will be able to view a Summary of the information you've provided. Please read this Summary page carefully before submitting to ensure you haven't made any errors!

Once you've completed the registration application, your Project status will be visible at https://www.audiokinetic.com/customers/projects/

Once approved, your Project status will turn from Pending to Active. Your Project must be Active before you can start using Wwise!

Installing Wwise

We recommend not installing or setting up Wwise until you have acquired your non-commercial License Key.

Download the Wwise Launcher for Windows (https://www.audiokinetic.com/download/)

Install and run the Wwise Launcher.

Once installation is complete, open the Wwise Launcher and click the Wwise tab.

Click the 'Latest' button and select 'All' from the dropdown.

In the dropdown to the right, select 2019.1.4.7065 as your version. This is the version used by MechWarrior 5: Mercenaries. You **must** use the same version!

Confirm that your changes reflect the steps above, then click the 'Install' button.

The Wwise Launcher will start retrieving some information from the server. When that process is complete you will be presented with the following screen.
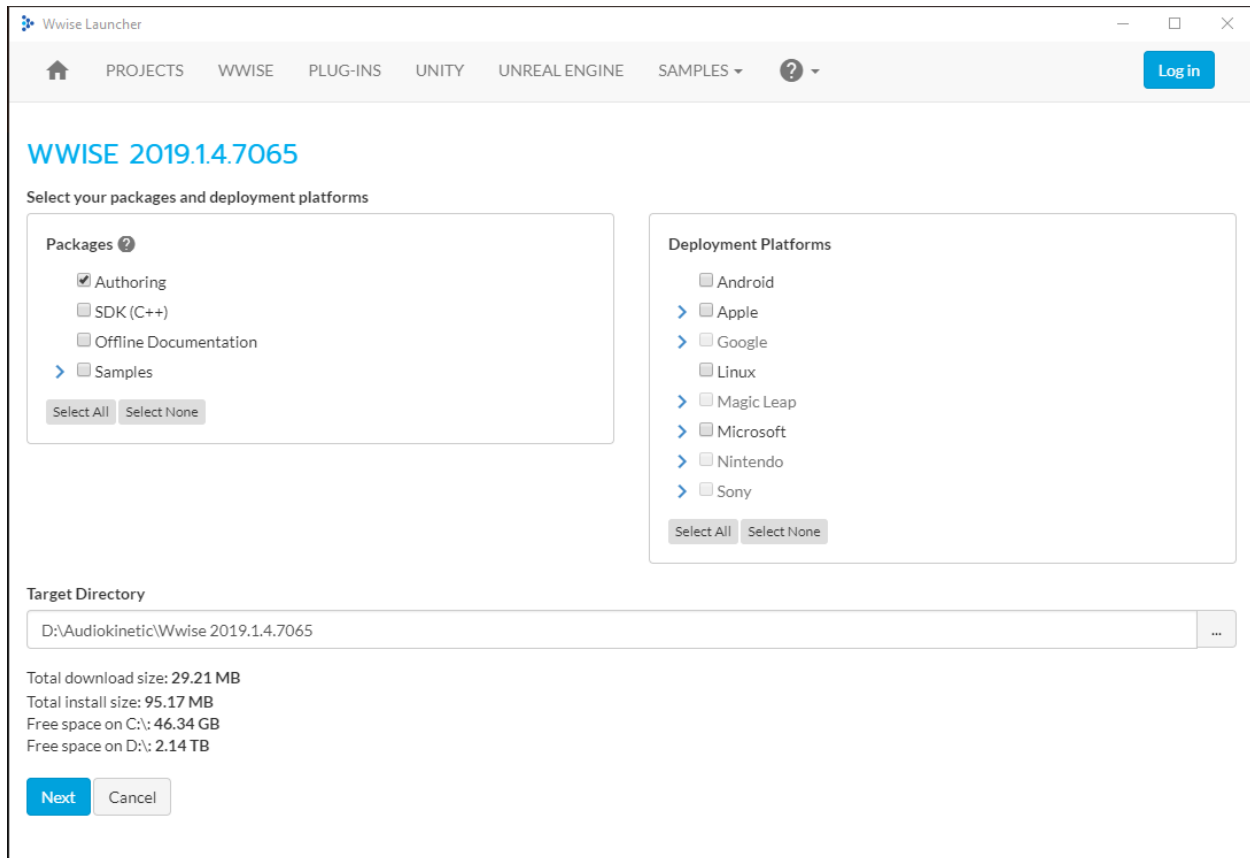
In the 'Packages' section on the left, ensure the 'Authoring' box is checked, and ensure the SDK (C++) box is **NOT** checked, as the SDK will not be required for using the MW5 Mod Editor.

If you wish, you can also select 'Offline Documentation' and 'Samples' to get additional resources for learning Wwise.

In the 'Deployment Platforms' section on the right, ensure none of the options are checked.

Specify a desired installation location in the 'Target Directory' field at the bottom. This location is where your Wwise Authoring Software will be installed. **THIS TARGET DIRECTORY SHOULD <u>NOT</u> BE THE LOCATION OF THE MW5MERCS EDITOR.**

When ready, click the 'Next' button.

You will be taken to a screen for choosing which Wwise Plug-Ins you wish to install.

From the list, only the following two Plug-Ins should be selected:

- Wwise Convolution

    - Version: 2019.1.4.7065

- McDSP ML1, McDSP Futzbox

    - Version: 2019.1.4.7065

As these Plug-Ins are required by MechWarrior 5: Mercenaries, your non-Commercial Wwise License will include licenses for these Plug-Ins.
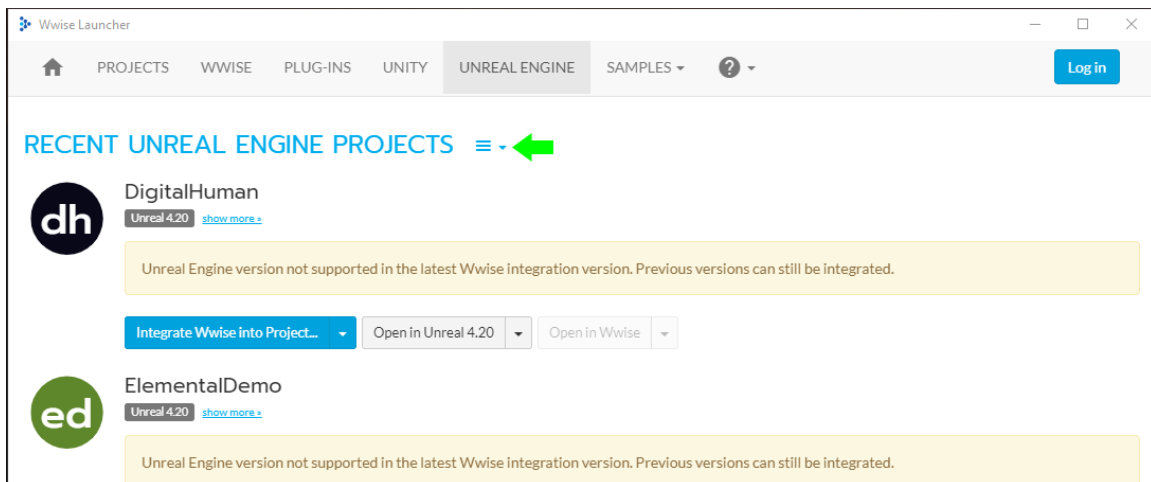
When ready, click the 'Install' button at the very bottom.

When the installation is complete you will now see an entry for Wwise version 2019.1.4.7065 in your Wwise tab.
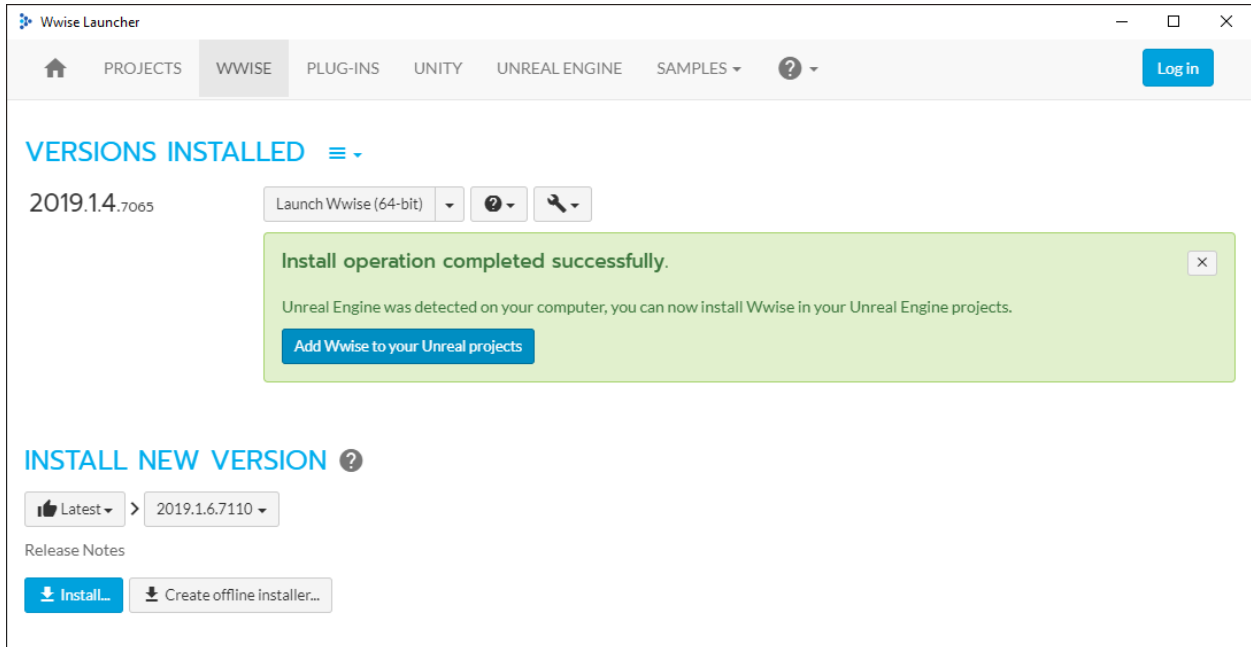
Click the 'Add Wwise to your Unreal Projects' button, or click the UNREAL ENGINE tab at the top of the Launcher.
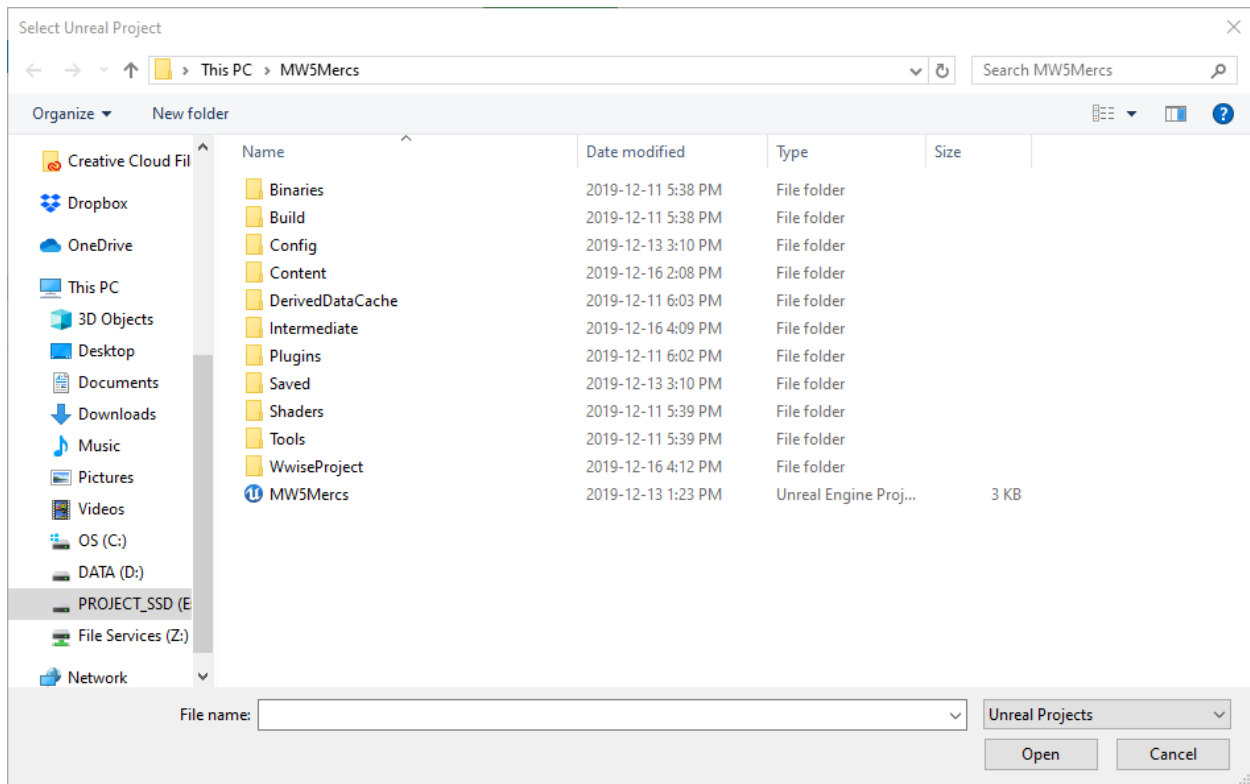
Click the icon next to RECENT UNREAL ENGINE PROJECTS

Select BROWSE FOR PROJECT... from the list of options

In the Explorer window that appears, navigate to <MW5 Mod Editor install location>\MechWarrior5Editor\MW5Mercs

Select the MW5Mercs project file and click Open

You will now see an entry for MW5Mercs in the list of Recent Unreal Engine Projects.

Do **NOT** click the 'Integrate Wwise into Project...' button or the 'Open in Unreal 4.21' button.

Instead, click the 'Open in Wwise 2019.1.4.7065' button to launch Wwise Authoring for MW5!

Alternatively, you can navigate to <MW5 Mod Editor install location>\MechWarrior5Editor\MW5Mercs\WwiseProject folder and open the file called ghost.uproj. This will also launch Wwise Authoring for MW5.

Getting Started

Once Wwise has been installed and your non-commercial License has been acquired, launch Wwise for MW5 either through the Wwise Launcher or directly through the ghost.uproj file mentioned above.

Wwise is a highly advanced and powerful tool for audio creation, and it will probably seem overwhelming at first. While a complete walkthrough of Wwise and its features is well beyond the scope of this guide, the following section will give you a basic rundown on the following tasks:

- Importing a sound file into the MW5 Wwise Project

- Replacing existing MW5 audio data with your own

- Generating the SoundBanks to propagate your audio changes to the MW5 Mod Editor

Adding your Non-Commercial License

Launch the MW5Mercs Wwise project (either through the Wwise Launcher or by opening the ghost.uproj file located in the <MW5 Mod Editor install location>\MechWarrior5Editor\MW5Mercs\WwiseProject folder.)

Before you start working, you'll want to apply your non-commercial License to the Project. To do so, select the 'License Manager' from the Projects dropdown in the toolbar (or hit CTRL+L on your keyboard).

When the License Manager panel appears you will see that no license is currently applied.
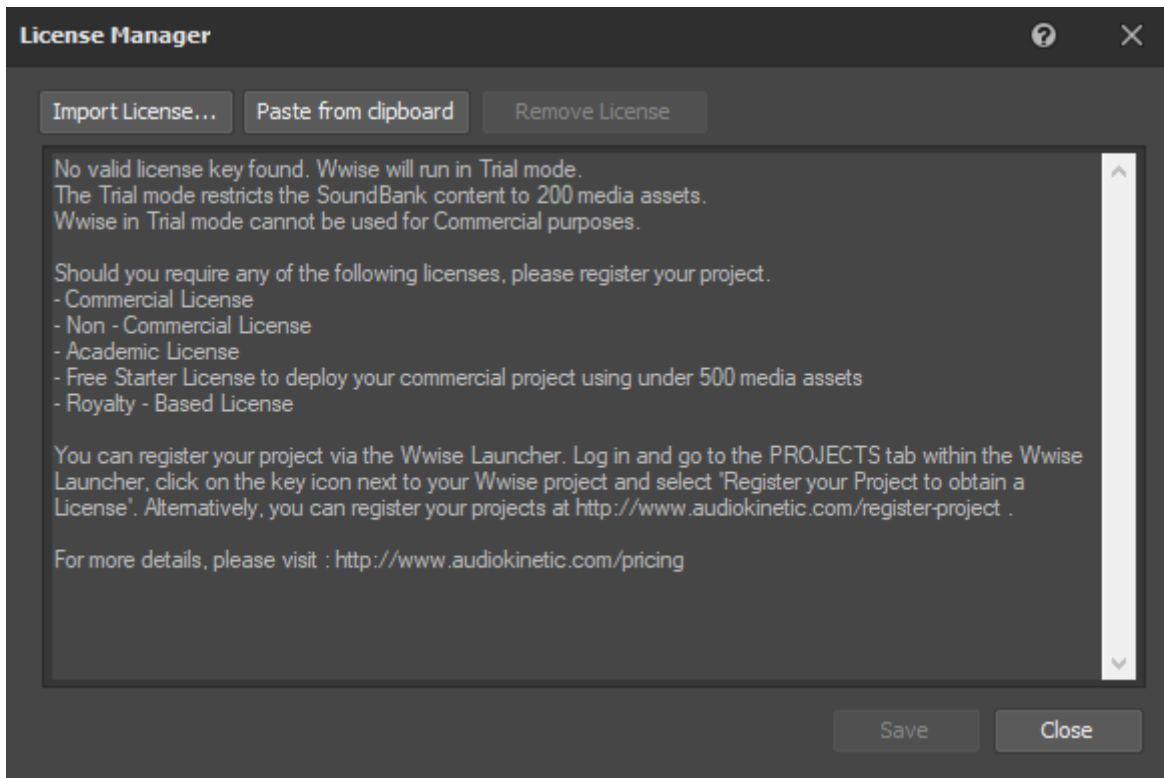
To get your License Key, navigate to https://www.audiokinetic.com/customers/projects/ and click the Licenses option on the left.

About your License

Note that you will need to get a new license key whenever your project licenses change.

Get license key   **View license details**

Wwise

Licenses		Expiration	Days Remaining	Platforms	
Academic	Level A License	2020-04-01	106		⊞
Academic	Maintenance	2020-04-01	106		⊞
Academic	Wwise	2020-04-01	106		⊞

Click Get License Key and copy the key string that appears.

In Wwise, click the Paste from Clipboard button in the License Manager.

The License Manager will update and display the details of your License.

License Manager   ? ✕

Import License...   Paste from clipboard   Remove License

Project ID:

Wwise for Windows (Academic/Non-Commercial): expires on 2020-04-01
Maintenance for Windows (Academic/Non-Commercial): expires on 2020-04-01
Level A License for Windows (Academic/Non-Commercial): expires on 2020-04-01
McDSP Bundle (ML1, FutzBox) for Windows (Academic/Non-Commercial): expires on 2020-04-01
Wwise Convolution Reverb for Windows (Academic/Non-Commercial): expires on 2020-04-01

Your Wwise project will be saved to embed the license.

Save   Cancel

Click Save. Your full non-commercial license has now been applied!

Important Note: Over time we may issue updates to the MW5 Mod Editor. These updates may result in your License information being removed from the MW5 Wwise Project. If this occurs, you'll simply need to reapply your License using the above steps.

Importing your sound file into the MW5 Wwise Project

The Project Explorer tab will be your primary method for viewing and selecting the various sound assets in MW5 that you wish to modify.

You specifically want to expand the main entry titled 'Actor-Mixer Hierarchy'. This is where all the individual sound 'actions' in MW5 are located, such as weapon fire effects, 'Mech movement sounds, and so on.
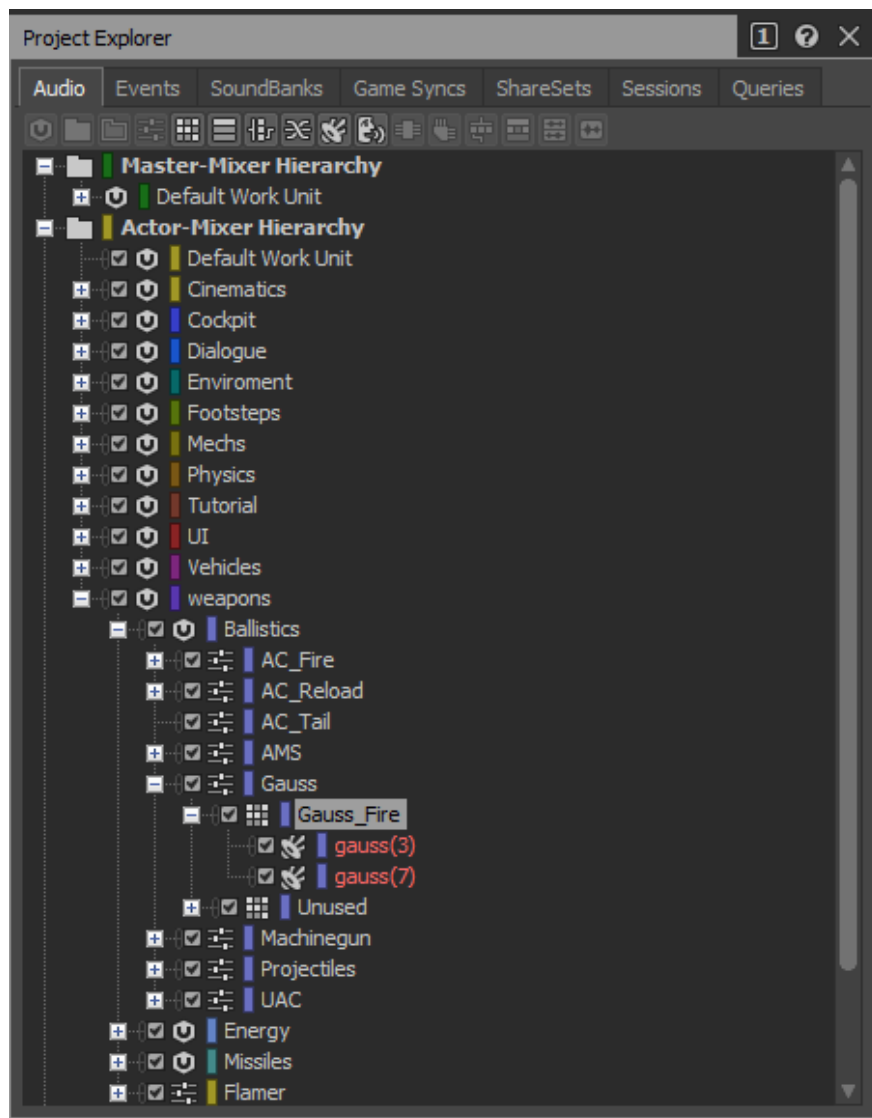
With 'Actor-Mixer Hierarchy' expanded you'll see a number of sub-categories, such as Cinematics, Cockpit, Mechs, and Weapons.

For now, let's just expand Weapons > Ballistics > Gauss > Gauss_Fire.

Here you'll see two entries:

- gauss(3)

- gauss(7)

We have two different Gauss SFX in the project by default. Every time a Gauss is fired, one of the two sounds is randomly chosen to be played.

In this example we'll be removing the second effect and replacing the first, leaving us with just one Gauss sound that plays every time it's fired.

First, let's delete the gauss(7) event by right-clicking it in the Project Explorer and selecting Delete.

Now let's import a new sound effect to use for gauss(3).

Right click the gauss(3) entry and select the Import Audio Files... option to bring up the Audio File Importer (or hit Shift+I on your keyboard).

Select the Create New Objects option from the dropdown next to Import Mode at the top left of the Audio File Importer window.

The Audio File Destination field is where your source files need to be located. They can be located in sub-folders of this location, but should not be located outside of it.

The Object Destination field will refer to the audio action for which you are trying to import a new source file. In this case, gauss(3).

Click the Add Files... button. Select your intended audio file from the Audio File Destination folder. If you receive any errors about a compatibility issue with your audio file you will need to resolve it by converting your audio file into a valid format.

Your selected file will now appear in the Audio File Importer window. Review the window then click the Import button at the bottom right.
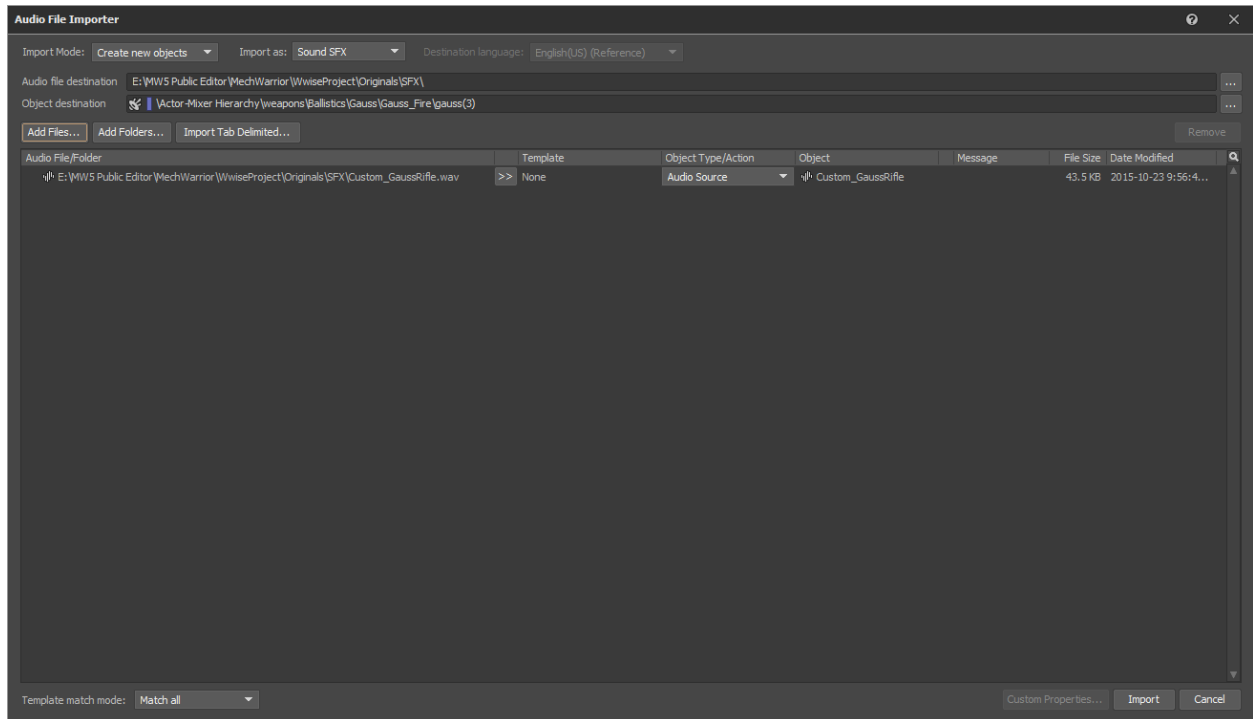
Returning to the main Wwise Project window you'll now see a new file in the details for gauss(3). In this example, we've added a file called Custom_GaussRifle.
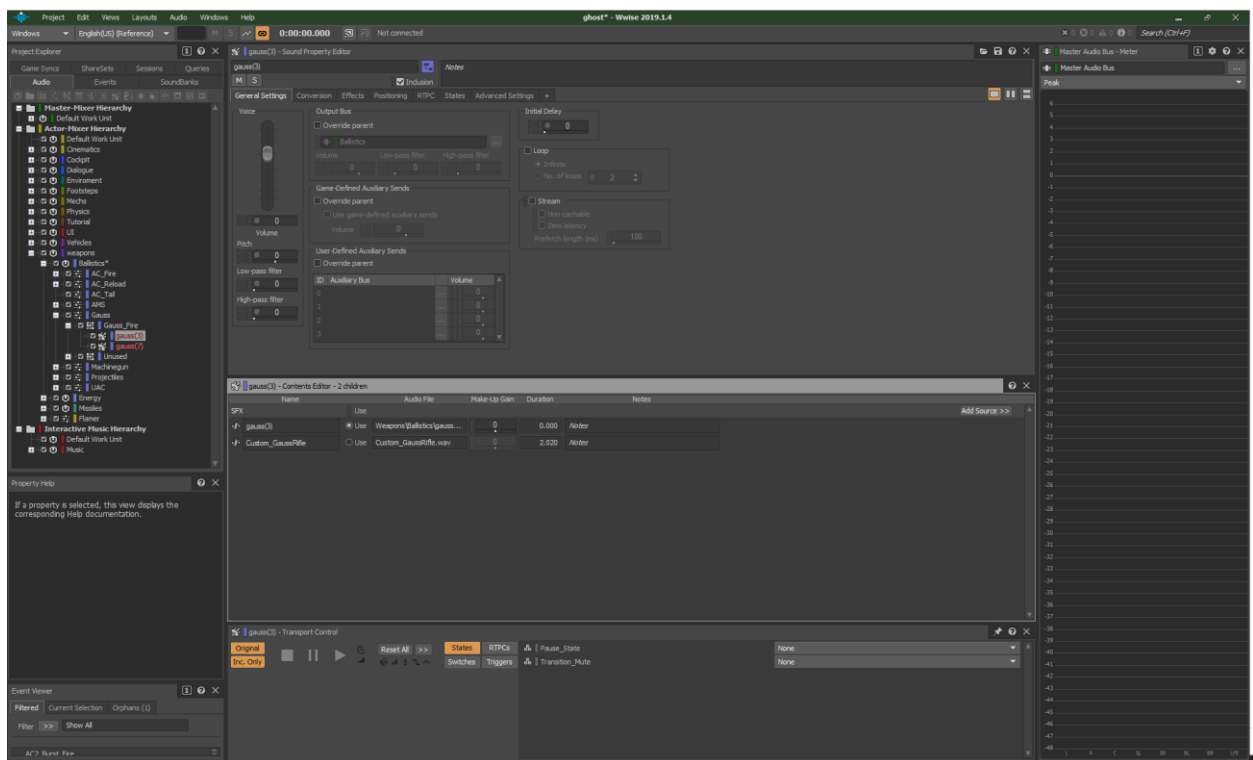

In the image above, note the 'Use' option is checked next to the original gauss(3) asset. To make the game use your new audio file, just click the Use option next to your new file.

When your audio file is selected, the Play button in the Transport Control panel below will light up, allowing you to preview the new audio file.
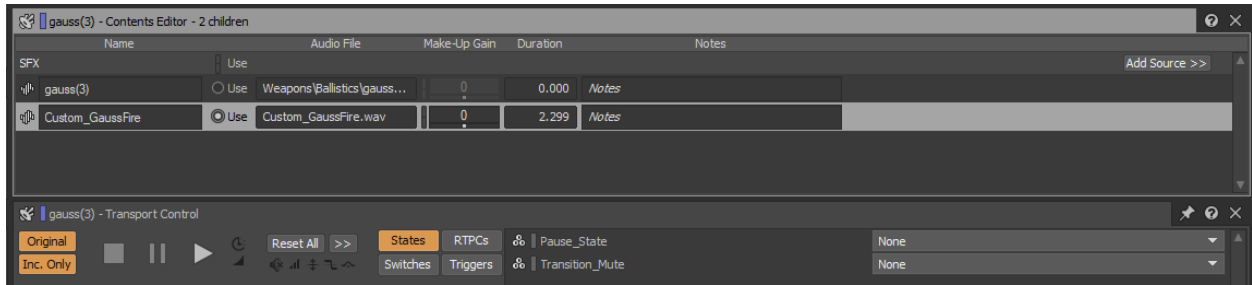
Now that you've added your first sound, hit Ctrl+S on your keyboard to save the Project (or select save from the Project option in the toolbar).

If you receive an error about a file being 'Read Only' when trying to save your Project, navigate to your <Mod Editor install location>\MW5Mercs folder. Right click the WwiseProject folder and select Properties.
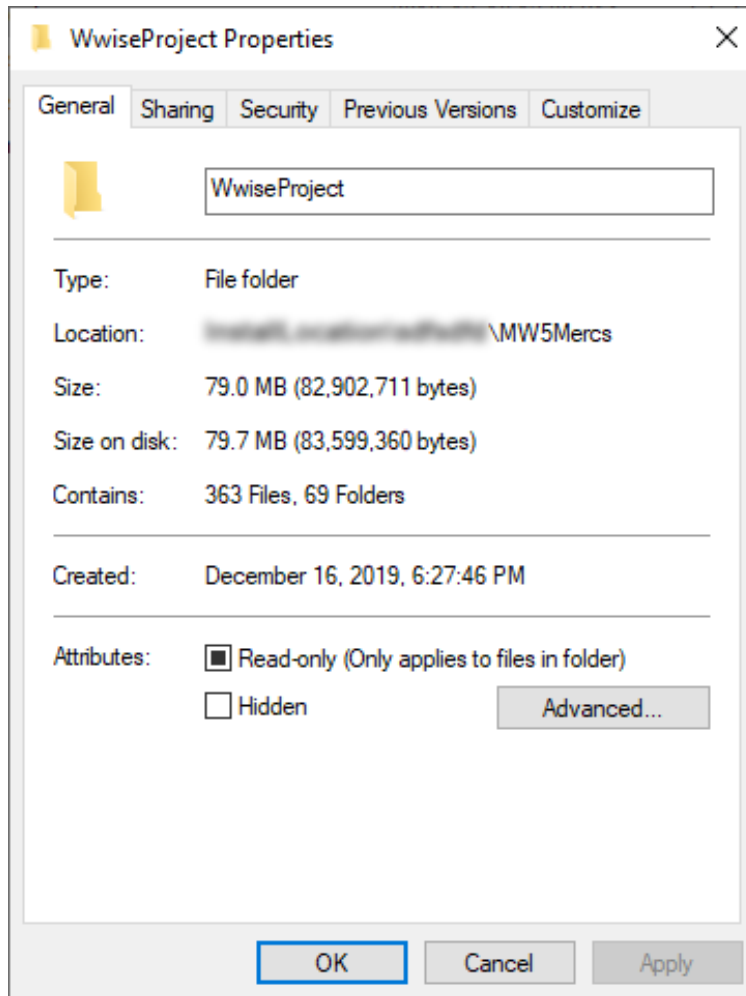
Uncheck the Read-only box and hit OK. In the confirmation window that appears, hit OK.

In Wwise, save the project again and it should save successfully.

With the new sound added and the Project saved, close Wwise and launch the MW5 Mod Editor.

Setting up the Editor with Wwise

Before going any further you'll need to make some changes to the MW5 Mod Editor to complete its linkage with your Wwise installation.

After launching the MW5 Mod Editor and letting the asset discovery complete, select Project Settings from the Edit dropdown in the toolbar.

Type Wwise into the search field.
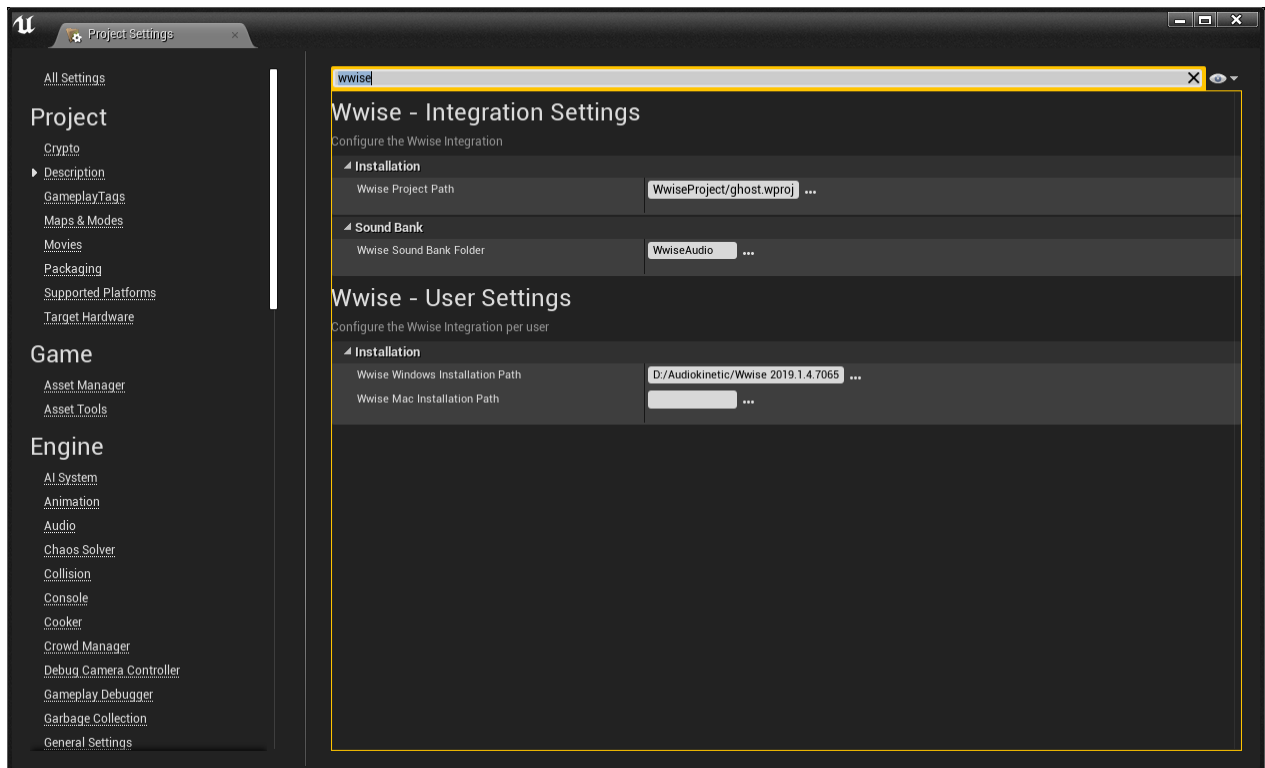
Click the ellipses (...) next to the Wwise Windows Installation Path. In the Explorer window that appears, navigate to the root folder of your Wwise installation, and click the 'Select Folder' button.
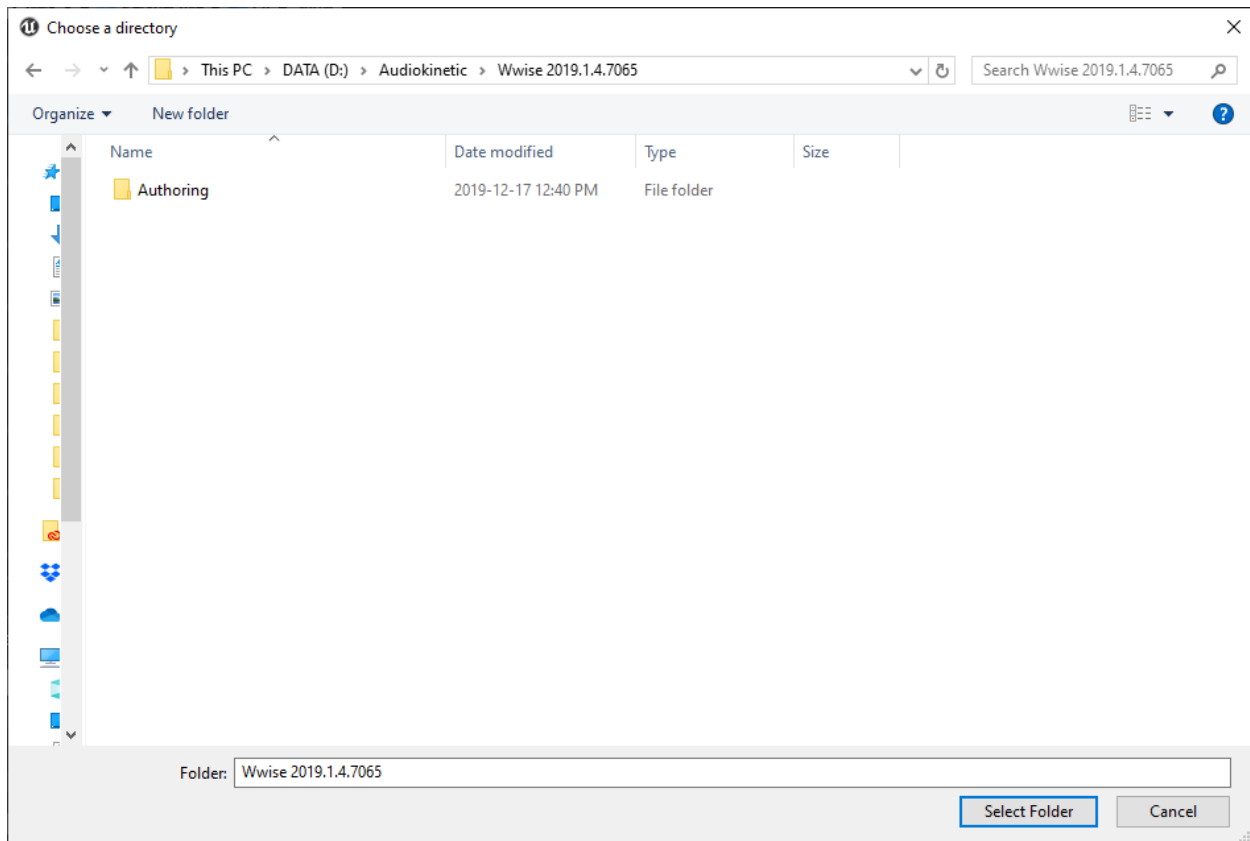
Now that you've completed this step, we need to regenerate and refresh the Weapons SoundBank, which contains your changes to the Gauss_Fire event.

Navigate to the Audio\Banks folder in the Editor Content Browser.

In this folder you'll see a number of Audiokinetic Bank assets. You may notice these correspond to the parent categories you saw in the Wwise project. In this example, the Weapons SoundBank contains the Gauss_Fire effect that we replaced.
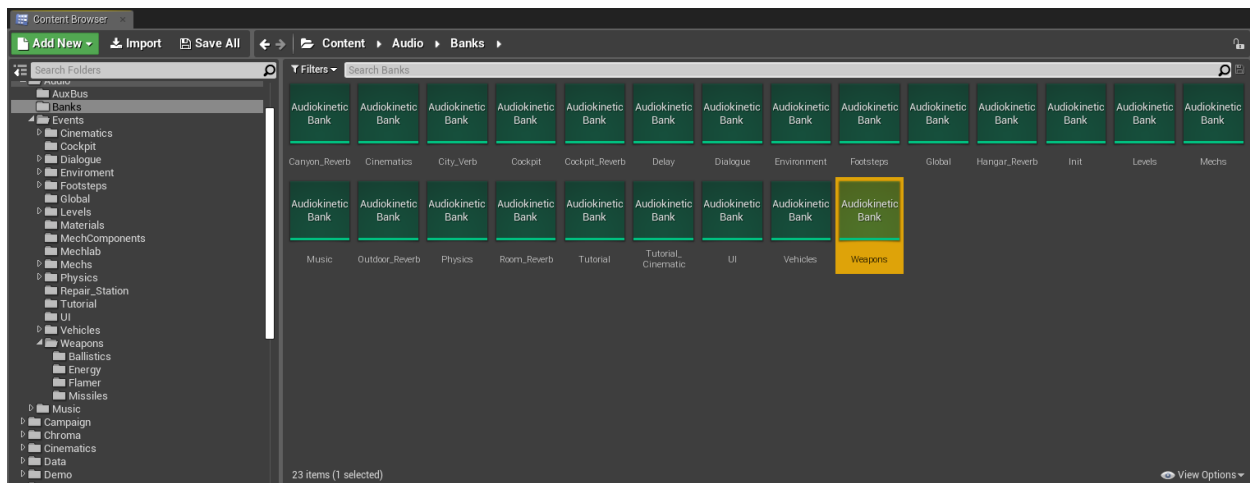
Right click the Weapons AudioKinetic Bank asset and select 'Generate Selected Soundbank...' from the top of the context menu.

In the window that appears, hit the 'Generate' button.

The Editor will begin the process of regenerating the Weapons SoundBank.

Important Note: The Editor will eventually report that the generation failed! This is normal, as it's simply stating that any assets for which you don't have source audio has failed to generate. In this example we've replaced the Gauss effect with our own source audio, and despite the reported generation failure, your Gauss change has been successfully generated!

Now, **right click the Weapons Audiokinetic Bank asset once again.** This time, select the Refresh All Banks option from the context menu.

Once complete, navigate to the Audio\Events\Weapons\Ballistics folder.

In this folder you'll see an Audiokinetic Event for the Gauss_Fire event. As you've modified this Event in Wwise with your new sound effect, regenerated the Weapons SoundBank, and refreshed all of the SoundBanks, the Gauss_Fire Audiokinetic Event in the Mod Editor will now use your audio when that event is played!
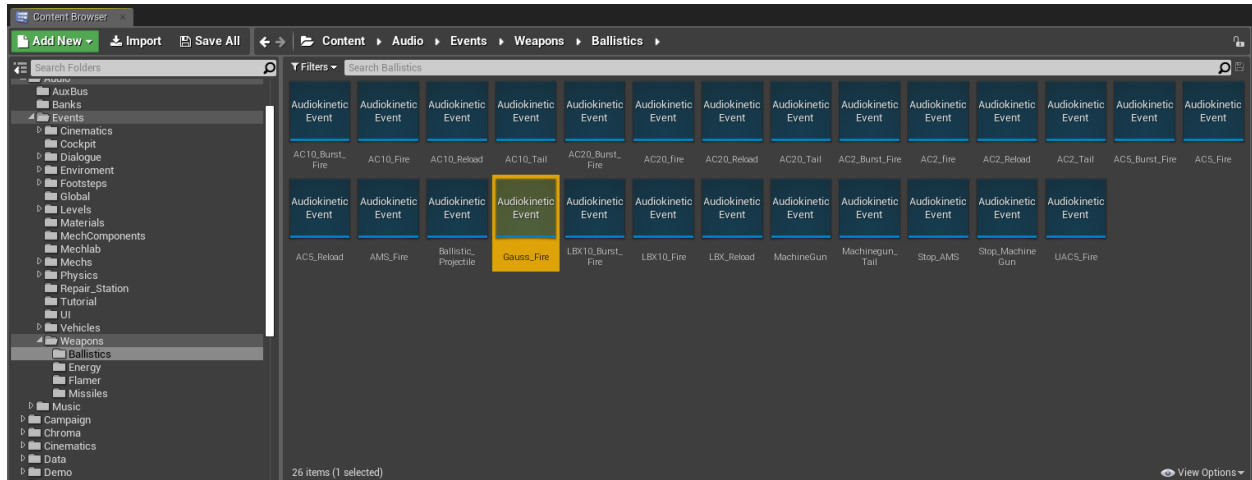
You can test your change by right-clicking the Gauss_Fire Audiokinetic Event and selecting the 'Play Event' option from the context menu. You should now hear your new Gauss effect! If you don't, run through the Wwise section of this guide again, paying careful attention to follow every step.

Important steps before packaging your Audio Mod

Packaging an Audio Mod is the same as packaging any other Mod, but there is one additional step required before you reach the packaging stage.

In order for your Audio Mod to be properly read by the game you need to pack some additional files along with it that aren't visible inside the Editor. For this process we'll need to open up a Windows Explorer window and copy those files into your Mod folder.

Open a Windows Explorer window and navigate to the \MW5Mercs\Content\WwiseAudio\Windows folder of your Mod Editor.

Sort by Date Modified to make things easier and you should see a few files that were recently updated from the SoundBank regeneration you performed in the previous section.

In this example we changed the Gauss fire sound, and all the files in the screenshot below need to be brought over to our Mod folder in order for our Gauss Mod to work in-game.

These files don't need to be modified or worked on in the Mod Editor. You just need to paste them directly into the MW5Mercs\Plugins\<ModName>\ModOverride\WwiseAudio\Windows folder of your Mod Editor. Note that after \ModOverride, the path matches that of the \WwiseAudio\Windows folder from which we copied these assets.

So for this example, with our Gauss audio Mod, our folder looks like this:
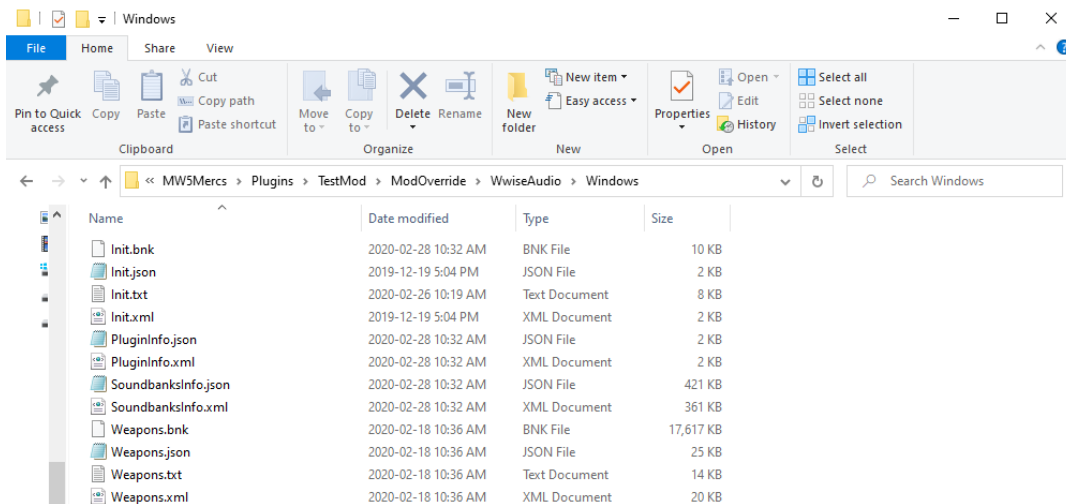
Now that those files exist in our Mod folder we can proceed with packaging the Mod through the Manage Mod interface in the Mod Editor (the process for which is outlined in a previous section). When you run the packaging process through the Manage Mod interface of the Mod Editor it will automatically package this WwiseAudio folder for you.